# A Multilingual Named Entity Recognition System Using Boosting and C4.5 Decision Tree Learning Algorithms

György Szarvas[1], Richárd Farkas[2], and András Kocsor[2]

[1] University of Szeged, Department of Informatics,
6720 Szeged, Árpád tér 2., Hungary
[2] MTA-SZTE, Research Group on Artificial Intelligence,
6720 Szeged, Aradi Vértanúk tere 1., Hungary
{rfarkas, szarvas, kocsor}@inf.u-szeged.hu

**Abstract.** In this paper we introduce a multilingual Named Entity Recognition (NER) system that uses statistical modeling techniques. The system identifies and classifies NEs in the Hungarian and English languages by applying AdaBoostM1 and the C4.5 decision tree learning algorithm. We focused on building as large a feature set as possible, and used a split and recombine technique to fully exploit its potentials. This methodology provided an opportunity to train several independent decision tree classifiers based on different subsets of features and combine their decisions in a majority voting scheme. The corpus made for the CoNLL 2003 conference and a segment of Szeged Corpus was used for training and validation purposes. Both of them consist entirely of newswire articles. Our system remains portable across languages without requiring any major modification and slightly outperforms the best system of CoNLL 2003, and achieved a 94.77% F measure for Hungarian. The real value of our approach lies in its different basis compared to other top performing models for English, which makes our system extremely successful when used in combination with CoNLL modells.

**Keywords:** Named Entity Recognition, NER, Boosting, C4.5, decision tree, voting, machine learning.

## 1 Introduction

The identification and classification of proper nouns in plain text is of key importance in numerous natural language processing applications. In Information Extraction systems proper names generally carry important information about the text itself, and thus are targets for extraction and Machine Translation. These have to handle proper nouns and other sort of words in a different way due to the specific translation rules that apply to them. These two topics are in the focus of our research.

### 1.1 Related Work

Research and development efforts in the last few years have focused on other languages, domains or cross-language recognition. Hungarian NER fits into this trend quite well, due to the special agglutinative property of the language.

N. Lavrač, L. Todorovski, and K.P. Jantke (Eds.): DS 2006, LNAI 4265, pp. 267 – 278, 2006.

Machine learning methods have been applied to the NER problem with remarkable success. The most frequently applied techniques were the Maximum Entropy Model, Hidden Markov Models (CoNLL-2003) and Support Vector Machines (JNLPBA-2004, [10]).

We use AdaBoostM1 and C4.5 learning techniques which have an inherently different theoretical background from the machine learning algorithms that have been used most frequently for NER (like Maximum Entropy Models, Support Vector Classifiers, etc.). The results of this paper prove that this can significantly improve classification accuracy in a model combination scheme. Another reason for using decision trees was that we needed a fast and efficient model to exploit the potentials of our large feature set.

There are some results on NER for the Hungarian language as well but all of them are based on expert rules [9], [12]. To our knowledge, no statistical models have yet been constructed for the Hungarian language.

### 1.2 Structure of the Paper

In the following section we will introduce the NER problem in general, along with the details of the English and Hungarian tasks performed and the evaluation methodology. In Section 3 we discuss the learning methods, the pre- and post-processing techniques we applied and the structure of our complex NER system. The experimental results are then presented in Section 4 along with a brief discussion, followed in Section 5 by some concluding remarks and suggestions for future work.

## 2 The NER Task

The identification of proper names can be regarded as a tagging problem where the aim is to assign the correct tag (label) to each token in a simple text. This classification determines whether the lexical unit in question is part of a proper noun phrase and if it is, which category it belongs to.

The NER task was introduced during the nineties as a part of the shared tasks in the Message Understanding Conferences (MUC) [4]. The goal of these conferences was the recognition of proper nouns (person, organization, location names), and other phrases denoting dates, time intervals, and measures in texts collected from English newspaper articles. The best systems [1] following the MUC task definition achieved outstanding accuracies (near 95% F measure).

Later, as a part of the Computational Natural Language Learning (CoNLL) conferences [15], a shared task dealt with the development of systems like this that work for multiple languages (first introduced in [5]) and were able to correctly identify a person, an organization and location names, along with other proper nouns treated as miscellaneous entities. The collection of texts consisted of newswire articles, in Spanish + Dutch and English + German, respectively. There are several differences between the CoNLL style task definition and the 1990s MUC approach that made NER a much harder problem:

- Multilinguality was introduced, thus systems had to perform well in more than one language without any major modification.

- The NE types that are very simple to identify (phrases denoting dates, time intervals, measures and so on.) were excluded from the CoNLL task. This way, systems were evaluated on the 4 most problematic classes out of the many used in MUCs.
- A more strict evaluation script was introduced that penalizes the misclassification of an inner part of a long phrase twice (one error for finding a wrong shorter phrase and another for the misclassified term).

These modifications made the NER task harder (the accuracy of the best performing systems [8] dropped below 89% for English) but more practical since real world applications like Information Extraction benefit from these types of NEs and by doing this (only whole phrases classified correctly contribute to other applications). In our studies we always followed the CoNLL style task definition and used the same evaluation script.

In accordance with the task definition of the CoNLL conferences we distinguish four classes of NEs, namely *person*, *location*, *organization names* and *miscellaneous entities*. This classification is not straightforward in many cases and a human annotator needs some background knowledge and additional information about the context to perform the task. Many proper nouns can denote entities of more than one class depending on the context, and occasionally a single phrase might fall into any of the four categories depending on the context (like "Ford", which can refer to a person, the company, an airport or the car type).

### 2.1 English NER

An NER system in English was trained and tested on a sub-corpus of the Reuters Corpus[1], consisting of newswire articles from 1996 provided by Reuters Inc. The data is available free of charge for research purposes and contains texts from diverse domains ranging from sports news to politics and the economy. The best result published in the CoNLL 2003 conference was an F measure of 88.76% obtained from the best individual model, and 90.3% for a hybrid model based on the majority voting of five participating systems.

### 2.2 Hungarian NER

To train and test our NER model on Hungarian texts, we decided to use a sub-corpus of the Szeged Treebank [6] which contains business news articles from 38 NewsML[2] topics ranging from acquisitions to stock market changes or the opening of new industrial plants. We annotated this collection of texts with NE labels that followed the current international standards as no other NE corpus of reasonable size is available for Hungarian. The data can be obtained free of charge for research purposes[3].

One major difference between Hungarian and English data is the domain specificity of the former corpus. The Hungarian texts we used consist of short newspaper articles from the domain of economy, and thus the *organization* class dominates the

[1] http://www.reuters.com/researchandstandards/
[2] See http://www.newsml.org/pages/docu_main.php for details.
[3] http://www.inf.u-szeged.hu/~hlt/index.html

other three in frequency. This difference undoubtedly makes NER an easier problem on the Hungarian text, while the special characteristics of the Hungarian language (compared to English) like agglutinativity or free word order usually makes NLP tasks in Hungarian very difficult. Thus it is hard to compare the results for Hungarian with other languages but achieving similar results in English (the language for which the best results have been reported so far) is still a remarkable feature.

The annotation procedure of the corpus consisted of several phases where two independent annotators tagged the data and discussed problematic cases later on. In the final phase all the entities that showed some kind of similarity to one that was judged inconsistent were collected together from the corpus for a review by the two annotators and the chief annotator. The resulting corpus had an inter-annotator agreement rate of 99.89% and 99.77% compared to the annotations made by the two linguists on their own [14].

This corpus then is completely equivalent to other corpuses used on the CoNLL-2002 and CoNLL-2003 conferences, both in format and annotation style (the same classes are labeled). We hope that this will make both cross-language comparison and the use of the corpus in developing NER systems more straightforward.

No independent results on Hungarian NER using this corpus have yet been published. The results here are compared to our previous results, which are the best that have been published so far [7].

### 2.3 Evaluation Methodology

To make our results easier to compare with those given in the literature, we employed the same evaluation script that was used during the CoNLL conference shared tasks for entity recognition[4]. This script calculates Precision, Recall and F value scores by analyzing the text at the phrase level. This way evaluation is very strict as it can penalize single mistakes in longer entity phrases two times.

It is worth mentioning that this kind of evaluation places a burden on the learning algorithms as they usually optimize their models based on a different accuracy measure. Fitting this evaluation into the learning phase is not straightforward because of some undesired properties of the formula that can adversely affect the optimization process.

## 3 Complex NER Model

We regard the NER problem as essentially a classification of separate tokens. We believe that this approach is competitive with the – theoretically more suitable – sequence tracking algorithms (like Hidden Markov Models, Maximum Entropy approaches or Conditional Random Fields) and we could choose a decision tree which requires less computation time and thus enables us for example to use an enormous feature set. Of course our model takes into account the relationship between consecutive words as well through a window with appropriate window size.

[4] The evaluation script can be downloaded from the CoNLL conference web site.

To solve classification problems effectively it is worth applying various types of classification methods, both separately[5] and in combination. The success of hybrid methods lies in tackling the problem from several angles, so algorithms of inherently different theoretical bases are good subjects for voting and for other combination schemes. Feature space construction and the proper pre-processing of data also have a marked impact on system performance. In our experiments we incorporated all these principles into a complex statistical NER model.

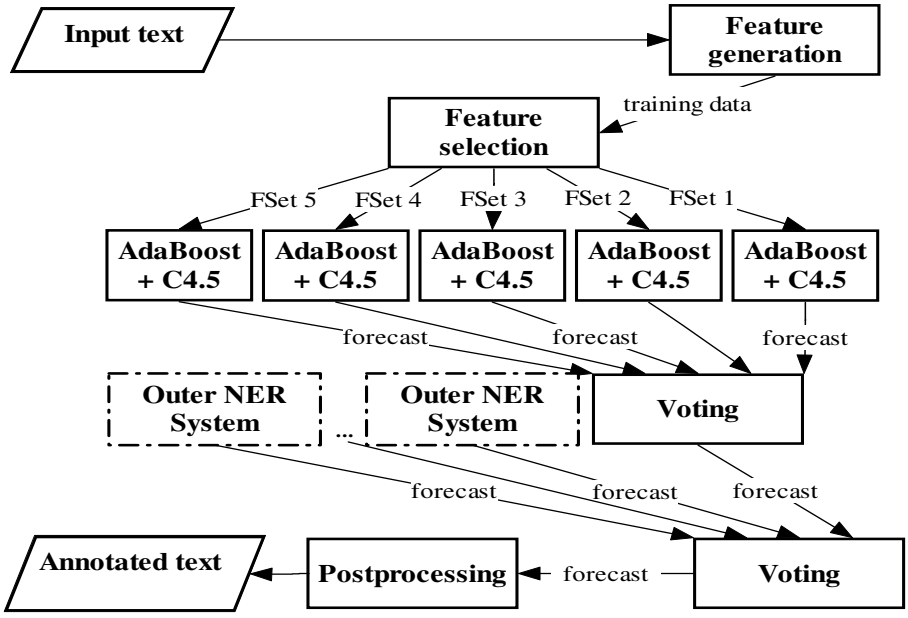


**Fig. 1.** Outline of the structure of our NER model. The result of our model working alone is discussed for English and Hungarian, along with the results of a voting system for English treated as a hybrid model. We used our model in combination with the two top performing CoNLL systems.

The building blocks of our system are shown in Figure 1. We expected our model would perform well in combination with other popular models (noted as "Outer NER System" in Figure 1) like the Maximum Entropy approach, Hidden Markov Model or Support Vector Classifiers. Our results on the English dataset where outputs of such systems were available justify this expectation.

### 3.1 Feature Set

**Initial features.** We employed a very rich feature set for our word-level classification model, describing the characteristics of the word itself along with its actual

[5] We investigated several algorithms but because of the lack of space we present here only the best performing one. For details of our past experiments, please see [7].

context (a moving window of size four). Our features fell into the following major categories:

- gazetteers of *unambiguous NEs* from the train data: we used the NE phrases which occur more than five in the train texts and got the same label more than 90 percent of the cases,
- *dictionaries* of first names, company types, sport teams, denominators of locations (mountains, city) and so on: we collected 12 English specific lists from the Internet and 4 additional to the Hungarian problem,
- *orthographical features*: capitalization, word length, common bit information about the word form (contains a digit or not, has uppercase character inside the word, and so on). We collected the most characteristic character level bi/trigrams from the train texts assigned to each NE class,
- *frequency information*: frequency of the token, the ratio of the token's capitalized and lowercase occurrences, the ratio of capitalized and sentence beginning frequencies of the token,
- *phrasal information*: chunk codes and forecasted class of few preceding words (we used online evaluation),
- *contextual information*: POS codes (we used codes generated by our POS tagger for Hungarian instead of the existing tags from the Szeged Treebank), sentence position, document zone (title or body), topic code, trigger words (the most frequent and unambiguous tokens in a window around the NEs) from the train text, is the word between quotes and so on.

The same features were used in the experiments on Hungarian texts and only the semantics of some feature varied (e.g. we used a different categorization of POS and chunk codes in Hungarian and the company type suffixes were different). Only the topic code and document zone features were omitted for Hungarian as all articles had the same topic (economics) and the titles of the articles were not capitalized like in the English dataset.

**Feature set splitting and recombination.** Using the above six groups of features we trained a decision tree for all possible subset of the groups (63 models), and of course not every subset describe the NER problem equally well. We used simple C4.5 trees here because their training are very fast and we assumed that the differences between single trees would not change significantly while boosting them. We evaluated these models on the CoNNL development set.

The 11 best performing models achieved very similar results; the others were far behind them. We decided to keep 5 of these models for CPU consumption reasons. We chose the five models – from the 11 - that showed the greatest average variety in features used (we did not omit a category because it achieved a slightly worse result).

We trained a classifier using each of these five feature sets (note that they are not disjunctive) and then recombined the resulting models in a voting scheme (which will be introduced later in detail). The same five sets of features (group of categories) were used in the experiments on Hungarian language.

### 3.2 Classifiers

Boosting [13] and C4.5 [11] are well known algorithms for those who are acquainted with pattern recognition. Boosting has been applied successfully to improve the performance of decision trees in several NLP tasks. A system that made use of AdaBoost and fixed depth decision trees [2] came first on the CoNLL-2002 conference shared task for Dutch and Spanish, but gave somewhat worse results for English and German (it was ranked fifth, and had an F measure of 85.0% for English) in 2003. We have not found any other competitive results for NER using decision tree classifiers and AdaBoost published so far.

As our results show, their combination can compete with state-of-the-art recognition systems solving the NER problem, as well as bring some improvement in classification accuracy and in preserving the superiority of decision tree learning when it comes to the CPU time used in training and evaluating a model. In our experiments we used the implementations available in the WEKA [16] library, an open-source data mining software written in Java.

**Boosting** was introduced by Shapire as a way of improving the performance of a weak learning algorithm. The algorithm generates a set of classifiers (of the same type) by applying bootstrapping on the original training data set and it makes a decision based on their votes. The final decision is made using a weighted voting schema for each classifier that is many times more accurate than the original model. 30 iterations of Boosting were performed on each model. Further iterations gave only slight improvements in the F measure (less than 0.05%), thus we decided to perform only 30 iterations in each experiment.

**C4.5** is based on the well-known ID3 tree learning algorithm, which is able to learn pre-defined discrete classes from labeled examples. Classification is done by axis-parallel hyperplanes, and hence learning is very fast. This makes C4.5 a good subject for boosting. We built decision trees that had at least 5 instances per leaf, and used pruning with subtree raising and a confidence factor of 0.33. These parameters were determined after the preliminary testing of some parameter settings and evaluating the decision trees on the development phase test set. A more thorough analysis of learning parameters will be performed in the near future.

### 3.3 Combination

There are several well known meta-learning algorithms in the literature that can lead to a 'better' model (in terms of classification accuracy) than those serving as a basis for it, or can significantly decrease the CPU time of the learning phase without loss of accuracy. In our study we chose to concentrate on improving the accuracy of the system.

The decision function we used to integrate the five hypotheses (learnt on different subsets of features) was the following: *if any three of the five learners' outputs coincided we accepted it as a joint prediction, with a forecasted 'O' label referring to a non-named entity class otherwise*. This cautious voting scheme is beneficial to system performance as a high rate of disagreement often means a poor prediction rate. For a CoNLL type evaluation it is better to make such mistakes that classifies an NE as non-named entity than place an NE in a wrong entity class (the latter detrimentally affects precision and recall, while the former only affects the recall of the system).

### 3.4 Post-processing Data

Several simple post-processing methods can bring about some improvement to system accuracy. Take, for instance, full person names which consist of first names and family names, which are easier to recognize than 'standalone' family names which refer to a person (e.g. "John Nash" or "Nash"). Here if we recognize a full name and encounter the family name later in the document we simply overwrite its label with a person name. This is a reasonable assumption that holds true in most cases.

Certain types of NEs rarely follow each other without any punctuation marks so if our term level classification model produces such an output we overwrite all class labels of this sequence with the label assigned to its head.

**Table 1.** Improvements of the post processing steps based on the previous step (percentile)

| | **Family names** | **Rare sequence filter** | **Acronym** |
|---|---|---|---|
| CoNLL Test | +0.63 | +0.59 | +0.70 |
| CoNLL Develop | +0.25 | +0.16 | +0.47 |

Acronym words are often easier to disambiguate in their longer phrase form, so if we find both in the same document we change the prediction given for the acronym when it does not coincide with the encountered longer form.

These simple post processing heuristics do not involve any learning or adaptation, but have been simply evaluated on the development dataset and found to be useful for both English and Hungarian – although their improvement on the Hungarian NER system was only marginal. Similar and other simple post-processing steps were performed in several NER systems (for example in [3], which came second in the CoNLL-2003 conference).

## 4 Results and Discussion

In this section we give a summary of our results and discuss the similarities and differences between Hungarian and English NER.

Tables 2 and 3 give a summary of the accuracies of the system elements for English and Hungarian texts. The effect of each element (which was built on the previous one) can be followed from top to bottom. In the first row one can see the performance of the baseline algorithm which selects complete unambiguous named entities appearing in the training data. The subsequent rows contain the results of the original feature set, the worst and the best models built on the five previously chosen feature sets, while the fourth row gives the performance of their combination. Here the feature set splitting procedure brought a significant (15-30%) error reduction. Finally the effect of the post processing steps can be seen in the last row.

Table 4 summarizes the F measure classification for each NE class. For English, location and person classes achieve the best accuracy, while organization is somewhat worse, and the miscellaneous class is much harder to identify. Our results for Hungarian indicate that organization can achieve an F measure comparable to location

**Table 2.** F measures of the recognition process for English

| | **Develop** | **Test** | **Error reduction (best)** |
|---|---|---|---|
| Baseline | | 59.61 | |
| Full FS | 87.17 | 84.81 | |
| Five models | 85.9-89.8 | 81.3-84.6 | |
| Voting | 91.40 | 86.90 | |
| Postproc | 92.28 | 89.02 | 2.32%[6] (6.08%) |
| Hybrid | 94.72 | 91.41 | 11.44% |

and person names (in the Hungarian data we had many more examples of organization names than those in the English corpus).

### 4.1 Results for English Texts

Our system got an F measure of 92.28% on the pre-defined development phase test set and 89.02% on the evaluation set (after a retraining which incorporated the development set into the training data) with the CoNLL evaluation script. This corresponds to a 2.32% error reduction relative to the best model known that was tested on the same data [8]. We should point out here that the system in [8] made use of the output of two externally trained NE taggers and thus the best standalone model in the system was [3]. When compared to it, it showed an error reduction of 6.08%.

Interestingly, we could improve both text sets at the same level (5 and 6.5 percentile), but while the feature set splitting procedure plays a key role in this improvement on the development set, post-processing helped the evaluation set more. This is because of the different characteristics of the sets.

Our algorithm was combined with the best two systems ([8], [3]) that were submitted to the CoNLL 2003 shared task[7], and performed significantly better than the best hybrid NER system reported in the shared task paper which employed the 5 best participating models (having a 91.41% F measure compared to 90.3%). This means a significant (11.44%) reduction in misclassified NEs. The successful applicability of our model in such a voting system is presumably due to ours having an inherently different theoretical background, which is usually beneficial to combination schemes. Our system uses Boosting and C4.5 decision tree learning, while the other two systems incorporate Robust Linear Classifier, Transformation-Based Learning, Maximum Entropy Classifier and Hidden Markov Model.

---

[6] The 4th row of Table 2 refers to the best individual system made by us and thus the error reduction was calculated against the best individual models, while the 5th row refers to the the hybrid model using our and two other CoNLL-2003 systems. The error reduction was calculated against the best hybrid system reported in the CoNLL-2003 shared task paper.

[7] Their output on the test set can be downloaded from the CoNLL homepage.

### 4.2 Results for Hungarian Texts

For Hungarian, the kind of results produced by inherently different but accurate systems are presently unavailable (thus the last voting step of Figure 1 is omitted in this case). However, our system gives fair results without the aid of the voting phase with outer systems. This is perhaps due to the domain specific nature of the input and makes NER a bit easier. The combined model which incorporated the predictions of the five AdaBoost+C4.5 models into a joint decision achieved an F measure of 94.77%.

**Table 3.** F measures of the recognition process for Hungarian

| | **Develop** | **Test** | **Error reduction (best)** |
|---|---|---|---|
| Baseline | | 70.99 | |
| Full FS | 95.21 | 92.77 | 3.08% |
| Five models | 90.3 – 94.7 | 88.1 – 93.7 | 0%-15.55% |
| Voting | 95.91 | 94.69 | 28.82% |
| Postproc | 96.20 | 94.77 | 29.89% |

These results are quite satisfactory if we take into account the fact that the results for English are by far the best known, and NLP tasks in Hungarian are many times more difficult to handle because Hungarian has many special (and from a statistical learning point of view, undesirable) characteristics.

**Table 4.** The per class F measures on the evaluation sets

| | **CoNLL individual** | **CoNLL hybrid** | **Hungarian** |
|---|---|---|---|
| LOC | 92.90 | 93.43 | 95.07 |
| MISC | 79.67 | 82.29 | 85.96 |
| ORG | 84.53 | 88.32 | 95.84 |
| PER | 93.55 | 96.27 | 94.67 |
| overall | 89.02 | 91.41 | 94.77 |

### 4.3 Discussion

Overall, then, we achieved some remarkably good results for NER; our systems can compete with the best known ones (and even perform slightly better on the CoNLL dataset). Being inherently different from those models that have been known to be successful in NER for English makes our system even more useful when it is combined with these competitive models in a decision committee. We should also mention here that our NER system remains portable across languages as long as

language specific resources are available; and it can be applied successfully to languages with very different characteristics.

For English our standalone model using AdaBoost and C4.5 with majority voting slightly outperforms other systems described in literature, although we should say that this difference is not significant. In spite of this, in experiments our system achieved a significant increase in prediction accuracy in combination with other competitive models.

## 5 Conclusions and Future Work

Our first conclusion here is that the building and testing of new or less frequently applied algorithms is always worth doing, since they can have a positive effect when combined with popular models. We consider the fact that we managed to build a competitive model based on a different theoretical background as the main reason for the significant (11.44%) decrease in misclassified NE phrases compared to the best hybrid system known.

Second, having a rich feature representation of the problem (which permits a feature set split and recombine procedure) often turns out to be just as important as the choice of the learning method.

Thirdly, our results demonstrate that combining well-known *general* machine learning methods (C4.5, Boosting, Feature Selection, Majority Voting) and *problem-specific* techniques (large feature set, post processing) into a complex system works well for NE recognition. What is more, this works well for different languages without the need for modifying the model itself, hence this task can be solved efficiently and in way that is language independent.

There are of course many ways in which our NER system could be improved. Perhaps the two most obvious ones are to implement those more popular models that we make use of majority voting (which is beneficial for the Hungarian model) and also to enlarge the size and improve the quality of our training data (the English dataset may contain some annotation errors and inconsistencies). This is what we plan to do in the near future.

## References

1. Daniel M. Bikel, Richard L. Schwartz, and Ralph M. Weischedel: An algorithm that learns what's in a name. Machine Learning, 34 -1-3 (1999) 211--231
2. Xavier Carreras, Lluís Márques and Lluís Padró: Named Entity Extraction using AdaBoost In: Proceedings of CoNLL-2002, Taipei, Taiwan, (2002) 167-170
3. Hai L. Chieu and Hwee T. Ng.: Named Entity Recognition with a Maximum Entropy Approach. Proceedings of CoNLL-2003 (2003) 160-163
4. Nancy Chinchor.: MUC-7 Named Entity Task Definition, in Proceedings of Seventh Message Understanding Conference (1998)
5. Silviu Cucerzan and Daniel Yarowsky: Language-independent named entity recognition combining morphological and contextual evidence. Proceedings of Joint SIGDAT Conf. on EMNLP/VLC (1999)

6. Dóra Csendes, János Csirik and Tibor Gyimóthy: The Szeged Corpus: A POS tagged and Syntactically Annotated Hungarian Natural Language Corpus. Proceedings of TSD 2004, vol. 3206 (2004) 41-49.
7. Farkas Richárd, Szarvas György, Kocsor András: Named Entity Recognition for Hungarian using various Machine Learning Algorithms accepted for publication in Acta Cybernetica (http://www.inf.u-szeged.hu/~rfarkas/ACTA2006_hun_namedentity.pdf)
8. Radu Florian, Abe Ittycheriah, Hongyan Jing and Tong Zhang: Named Entity Recognition through Classifier Combination. Proceedings of CoNLL-2003 (.2003) 168-171.
9. Kata Gábor, Enikő Héja, Ágnes Mészáros, Bálint Sass: Nyílt tokenosztályok reprezentációjának technológiája. IKTA-00037/2002, Budapest, Hungary (2002)
10. Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi and Nigel Collier: Introduction to the Bio-Entity Task at JNLPBA. Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (2004)
11. Ross Quinlan: C4.5: Programs for machine learning, Morgan Kaufmann (1993)
12. Gábor Prószéky: Syntax As Meta-Morphology. Proceedings of COLING-96, Vol.2 (1996) 1123–1126
13. Rob E. Shapire: The Strength of Weak Learnability. Machine Learnings, Vol. 5 (1990) 197-227
14. György Szarvas, Richárd Farkas, László Felföldi, András Kocsor, János Csirik: A highly accurate Named Entity corpus for Hungarian, Proceedings of International Conference on Language Resources and Evaluation (2006)
15. Erik F. Tjong Kim Sang, and Fien De Meulder: Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition, Proceedings of CoNLL-2003 (2003)
16. Ian H. Witten and Eibe Frank: Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco (2005)