Combining Data-Driven Systems for Improving Named Entity Recognition

Zornitsa Kozareva, Oscar Ferrández, Andres Montoyo,
Rafael Muñoz, and Armando Suárez

Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante
{zkozareva,ofe,montoyo,rafael,armando}@dlsi.ua.es

Abstract. The increasing flow of digital information requires the extraction, filtering and classification of pertinent information from large volumes of texts. An important preprocessing tool of these tasks consists of name entities recognition, which corresponds to a Name Entity Recognition (NER) task. In this paper we propose a completely automatic NER which involves identification of proper names in texts, and classification into a set of predefined categories of interest as Person names, Organizations (companies, government organizations, committees, etc.) and Locations (cities, countries, rivers, etc). We examined the differences in language models learned by different data-driven systems performing the same NLP tasks and how they can be exploited to yield a higher accuracy than the best individual system. Three NE classifiers (Hidden Markov Models, Maximum Entropy and Memory-based learner) are trained on the same corpus data and after comparison their outputs are combined using voting strategy. Results are encouraging since 98.5% accuracy for recognition and 84.94% accuracy for classification of NE for Spanish language were achieved.

1 Introduction

The vision of the information society as a global digital community is fast becoming a reality. Progress is being driven by innovation in business and technology, and the convergence of computing, telecommunications and information systems. Access to knowledge resources in the information society is vital to both our professional and personal development. However, access alone is not enough. We need to be able to select, classify, assimilate, retrieval, filter and exploit this information, in order to enrich our collective and individual knowledge and skills. This is a key area of application for language technologies. The approach taken in this area is to develop advanced applications characterized by more intuitive natural language interfaces and content-based information analysis, extraction and filtering. Natural Language Processing (NLP) is crucial in solving these tasks. In concrete, Name Entity Recognition (NER) has emerged as an important preprocessing tool for many NLP applications as Information Extraction, Information Retrieval and other text processing applications. NER

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 80–90, 2005.

involves processing a text and identifying certain occurrences of words or expressions as belonging to a particular category of Named Entities (NEs) as person, organization, location, etc.

This paper describes a multiple voting method that effectively combines strong classifiers such as Hidden Markov Models, Maximum Entropy and Memory-based learner for the NE recognition and classification tasks for Spanish texts. Two different approaches have been developed by the researchers in order to solve the Named Entity Recognition task. The former approach is based on Machine Learning methods, such us Hidden Markov's Models, Maximum Entropy, Support Vector Machine or Memory-based. This approach uses a set of features providing information about the context (previous and posterior words), orthographic features (capital letter, etc.), semantic information, morphological information, etc. in order to provide statistical classification. The latter approach is based on Knowledge-based techniques. This approach uses a set of rules to implement a specific grammar for named entity and set of databases or gazetteer to look for specific words like names of people or locations. List of names or gazetteer can be also used in future for machine learning method.

Different systems have been developed for each approach, we emphasize two conferences: CoNLL[1] and ACE[2], and several systems achieving good scores. We point out two knowledge-based systems like Maynard et al. [6], Arevalo et al. [1] and several machine learning systems like Carreras et al. [2], Mayfield et al. [5], Florian et al. [4], etc.

The organization of this paper is as following: After this introduction, the features used by our classifiers are listed in Section 2; the sheer classifiers are detailed in Section 3. The different experiments and obtained results are examined and discussed in Section 4. The voting strategy we used is in Section 5 and finally we conclude (Section 6) with a summary of the most important observations and future work.

2 Description of Features

The Maximum Entropy and Memory-based learning classifiers we used for the NE tasks (detection and classification) utilize the identical features described below. In contrast HMM doesn't take any features because it depends on the probability of the NE and the tag associated with it. To gain better results we studied different feature combinations from the original set.

2.1 Features for NE Detection

For NE detection, we use the well-known BIO model, where a tag shows that a word is at the beginning of a NE (B), inside a NE (I) or outside a NE (O). For the sentence "Juan Carlos está esperando", the following tags have been

[1] http://cnts.uia.ac.be/conll2002/ner/
[2] http://www.nist.gov/speech/tests/ace/

- **a**: anchor word (e.g. the word to be classified)
- ***c[1-6]***: context of the word at position ±1, ±2, ±3
- ***C[1-7]***: capitalization of the word at position 0, ±1, ±2, ±3
- ***d[1-3]***: word +1,+2,+3 in dictionary of entities
- ***p***: position of anchor word in the sentence

Fig. 1. Features for NE detection.

associated, "B I O O O", where *Juan* starts a named entity; *Carlos* continues this entity and neither *está* nor *esperando* or the full stop are part of a NE.

Our BIO model uses a set composed of 18 features as described in Figure 1. They represent words, position in the sentence and entity triggers for each NE.

2.2 Features for NE Classification

The tags used for NE classification are PER, LOC, ORG and MISC as defined by CoNLL-2002 task. Their detection is possible by the help of the first seven features used by out BIO model (e.g. a, c[1-6], p) and the additional set described below in Figure 2. In Section 4 several experiments were made by shortening the original set into one containing the most informative ones and their influence upon system's performance is discussed.

- ***eP***: entity is trigger PER
- ***eL***: entity is trigger LOC
- ***eO***: entity is trigger ORG
- ***eM***: entity is trigger MISC
- ***tP***: word ±1 is trigger PER
- ***tL***: word ±1 is trigger LOC
- ***tO***: word ±1 is trigger ORG
- ***gP***: part of NE is in database or gazzetters for PER
- ***gL***: part of NE is in database or gazzetters for LOC
- ***gO***: part of NE is in database or gazzetters for ORG
- ***wP***: whole entity is PER
- ***wL***: whole entity is LOC
- ***wO***: whole entity is ORG
- ***NoE***: whole entity is not in the defined three classes
- ***f***: first word of the entity
- ***s***: second word of the entity

Fig. 2. Features for NE classification.

3 Classification Methods

We have used three classification methods, in concrete Memory-based learner, Maximum Entropy and HMM for the NE detection and classification tasks. Next subsections describe each method individually.

3.1 Memory-Based Learner

Memory-based learning is a supervised inductive learning algorithm for learning classification tasks. It treats a set of training instances as points in a multi-

dimensional feature space, and stores them as such in an instance base in memory. Test instances are classified by matching them to all instances in memory, and by calculating with each match the distance, given by a distance function between the new instance x and each of the n memory instances $y_1 \ldots_n$. Classification in memory-based learning is performed by the $k - NN$ algorithm that searches for the k 'nearest neighbours' among the memory instances according to the distance function. The majority class of the k nearest neighbors then determines the class of the new instance x. [3]. The memory-based software package we used is called Timbl [3]. Its default learning algorithm, instance-based learning with information gain weighting (IB1IG) was applied.

3.2 Maximum Entropy

The maximum entropy framework estimates probabilities based on the principle of making as few assumptions as possible, other than the constraints imposed. The probability distribution that satisfies the above property is the one with the highest entropy [7]. A classifier obtained by means of a ME technique consists of a set of parameters or coefficients which are estimated using an optimization procedure. Each coefficient is associated with one feature observed in the training data. The main purpose is to obtain the probability distribution that maximizes the entropy. Some advantages of using the ME framework are that even knowledge-poor features may be applied accurately; the ME framework thus allows a virtually unrestricted ability to represent problem-specific knowledge in the form of features.

$$f(x,c) = \begin{cases} 1 & if \text{ c'=c\&cp(x)=true} \\ 0 & \text{otherwise} \end{cases} . \tag{1}$$

$$p(c|x) = \frac{1}{Z(x)} \prod_{i=1}^{K} \alpha_i^{f_i(x,c)} \tag{2}$$

The implementation of ME was done in C++[10] and the features used for testing are described in the section above. The implementation we used is a very basic one because no smoothing nor feature selection is performed.

3.3 Hidden Markov Models

Hidden Markov Models are stochastic finite-state automata with probabilities for the transitions between states and for the emission of symbols from states. The Viterbi algorithm is often used to find the most likely sequence of states for a given sequence of output symbols. In our case, let T be defined as set of all tags, and $\sum$ the set of all NEs. One is given a sequence of NEs $W = w_1 \ldots w_k \in \sum^*$, and is looking for a sequence of tags $T = t_1 \ldots t_k \in T^*$ that maximizes the conditional probability $p(T|W)$, hence we are looking for

$$arg \max_T p(T|W) = arg \max_T \frac{p(T)p(W|T)}{p(W)} \tag{3}$$

p(W) is independent of the chosen tag sequence, thus it is sufficient to find

$$arg \max_{T} p(T)p(W|T). \tag{4}$$

The toolkit we used is called ICOPOST[3] implemented for POS tagging purpose and adapted for NER [9].

4 Experiments and Discussion

Our NER system has two passages

1. detection : identification of sequence of words that make up the name of an entity.

2. classification : deciding to which category our previously recognized entity should belong.

The Spanish train and test data we used are part of the CoNLL-2002 [8] corpus. For training we had corpus containing 264715 tokens and 18794 entities and for testing we used Test-B corpus with 51533 tokens and 3558 entities. Scores were computed per NE class and the measures used are Precision (of the tags allocated by the system, how many were right), Recall (of the tags the system should have found, how many did it spot) and $F_{\beta=1}$(a combination of recall and precision). To calculate precision and recall for all tags in the system, Accuracy is used as Precision and Recall coincide (e.g. all NEs have a tag, and there is no case in which an entity has no class).

$$Precision = \frac{number\,of\,correct\,answers\,found\,by\,the\,system}{number\,of\,answers\,given\,by\,the\,system} \tag{5}$$

$$Recall = \frac{number\,of\,correct\,answers\,found\,by\,the\,system}{number\,of\,correct\,answers\,in\,the\,test\,corpus} \tag{6}$$

$$F_{\beta=1} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{7}$$

$$Accuracy = \frac{correctly\,classified\,tags}{total\,number\,of\,tags\,in\,the\,test\,corpus} \tag{8}$$

4.1 Recognition by BIO Model

During NE detection, Timbl and ME classifiers follow the BIO model described briefly in subsection 2.1, using the set of 18 features described in Figure 1 while HMM takes only the NE and the tag associated with it. Systems' performance can be observed in Table 1. For clearance of result calculation, we put in abbreviations the various tag combinations (B:B, B:I, B:O, etc.) and their values (column N). The first letter always points to the class the NE has in reality and the second one shows the class predicted by the classifier. For the case of B tags, B:I signifies that the NE is supposed to be B, but the classifier assigned an I tag

[3] http://acopost.sourceforge.net/

to it and for B:O the system put an O for an entity that is supposed to be B. The same holds for the other abbreviations. If a confusion matrix is built using the values in column "N" for each tag, the calculation of precision and recall can be obtained easily [3].

Table 1. BIO detection.

		Timbl(%)				HMM(%)				Maximum Entropy(%)			
		N	Prec.	Rec.	$F_{\beta=1}$	N	Prec.	Rec.	$F_{\beta=1}$	N	Prec.	Rec.	$F_{\beta=1}$
B	B:B	3344	93.59	93.99	93.79	3262	90.14	91.68	90.90	1060	85.42	29.79	44.18
	B:I	88				148				54			
	B:O	126				148				2444			
I	I:I	2263	91.18	86.37	88.71	2013	87.52	76.83	81.83	673	81.48	25.69	39.06
	I:B	157				246				127			
	I:O	200				361				1820			
O	O:O	45152	99.28	99.55	99.42	45105	98.88	99.45	99.17	45202	91.38	99.66	95.34
	O:B	72				111				54			
	O:I	131				139				99			
Accuracy		98.50				97.76				91.08			
Only B&I		precBI=93.16 recBI=90.76 $F_{\beta=1}BI$=92.27				precBI=89.12 recBI=85.38 $F_{\beta=1}BI$=87.21				precBI=83.84 recBI=28.05 $F_{\beta=1}BI$=42.04			

The coverage of tag O is high due to its frequent appearance, however its importance is not so significant as the one of B and I tags. For this reason we calculated separately system's precision, recall and F-measure for B and I tags together. The best score was obtained by the memory-based system Timbl with F-measure of 92.27%.

As a whole system's performance is calculated considering all BIO tags and the highest score of 98.50% Accuracy is achieved by Timbl. After error analysis we discovered that results can be improved with simple post-processing where in the case of I tag preceded by O tag we have to substitute it by B if the analyzed word starts with a capital letter and in the other case we simply have to put O (see the example in subsection 2.1). With post-processing Timbl raised its Accuracy to 98.61% , HMM to 97.96% and only ME lowered its score to 90.98%. We noticed that during ME's classifictaion occured errors in the O I I sequences which normally should be detected as B I I. The post-processing turns the O I I sequence into O B I which obviously is erroneous. First error is due to the classificator's assignement of an I tag after an O and the second one is comming from the post-processor's substitution of the first I tag in the O I I sequence by B. This errors lowered ME's performance.

4.2 Classification into Classes

After detection follows NE classification into LOC, MISC, ORG or PER class. Again to HMM model we passed the NE and its class. The achieved accuracy

is 74.37% and in Table 2 can be noticed that LOC, ORG and PER classes have good coverage while MISC has only 57.84% due to its generality.

Table 2. HMM Classifier.

Class	Prec.%	Rec.%	$F_{\beta=1}$%
LOC	80.02	73.16	76.43
MISC	56.46	59.29	57.84
ORG	77.60	77.71	77.66
PER	69.72	76.74	73.06
Accuracy	74.37%		

For the same task the other two systems used the set of features described in subsection 2.2. Initially we made experiments with a bigger set composed of 37 features extracted and collected from articles of people researching in the same area. They include the following 18 attributes: and the denoted in Figure 1 and 2 features: a, c[1-6], p, eP, eL, eO, eM, gP, gL, gO, wP, wL, wO, NoE.

- ***wtL[1-2]***: word ±1 is trigger LOC
- ***wtO[1-2]***: word ±1 is trigger ORG
- ***wtP[1-2]***: word ±1 is trigger PER
- ***wtL[1-2]***: word ±2 is trigger LOC
- ***wtO[1-2]***: word ±2 is trigger ORG
- ***wtP[1-2]***: word ±2 is trigger PER
- ***wtL[1-2]***: word ±3 is trigger LOC
- ***wtO[1-2]***: word ±3 is trigger ORG
- ***wtP[1-2]***: word ±3 is trigger PER

Fig. 3. Features for NE detection.

The obtained results from these attributes are in Table 3. Their accuracy is better than the one of HMM but still not satisfactory. Then we decided to investigate the most substantial ones, to remove the less significant and to include two more attributes. Thus our NE classification set became composed of 24 features as described above in subsection 2.2. Let us denote by A the set of features: a, c[1-6], p, eP, eL, eO, eM, tP, tL, tO, gP, gL, gO, wP, wL, wO, NoE, f and s.

Table 3. Timbl and ME using 37 features.

Class	*Timbl*			*Maximum entropy*		
	Prec.%	Rec.%	$F_{\beta=1}$%	Prec.%	Rec.%	$F_{\beta=1}$%
LOC	80.23	76.38	78.26	81.07	74.26	77.52
MISC	51.10	48.08	49.54	78.95	39.82	52.94
ORG	77.94	82.5	80.15	73.06	86.57	79.24
PER	83.17	82.04	82.60	78.64	78.64	78.64
Accuracy	77.26%			76.73%		

In Table 4 are shown the results of Timbl and ME using set A. Their accuracy has been higher than the one of HMM, however ME performed better than Timbl. The memory-based learner works by measuring distance among elements which made us select and test as a second step only the most relevant and informative features.

Table 4. Timbl and ME using set A.

Class	*Timbl*			*Maximum entropy*		
	Prec.%	Rec.%	$F_{\beta=1}$%	Prec.%	Rec.%	$F_{\beta=1}$%
LOC	82.02	80.81	81.41	88.25	81.09	84.52
MISC	64.83	61.95	63.35	85.65	58.11	69.24
ORG	81.61	84.93	83.23	80.66	90.29	85.20
PER	88.29	85.17	86.70	86.93	90.48	88.67
Accuracy	81.54%			84.46%		

Let denote by B the set with the most informative attributes which is a subset of our original set A: a, c[1], eP, gP, gL, gO, wP, wL, wO, NoE and f. Table 5 shows the results with the reduced set (e.g. B). It can be notices that Timbl increased its performance to 83.81% but ME lower it to 82.24%, because even knowledge-poor features are important and can be applied accurately to it. Timbl classified MISC class better with 0.35% than ME when using the whole feature set and got the higher coverage for this class among our classifiers.

Table 5. Timbl and ME using set B.

Class	*Timbl*			*Maximum entropy*		
	Prec. %	Rec. %	$F_{\beta=1}$ %	Prec. %	Rec. %	$F_{\beta=1}$ %
LOC	85.35	82.20	83.74	86.33	79.24	82.64
MISC	77.54	63.13	69.59	91.19	51.92	66.17
ORG	83.92	86.50	85.19	74.64	93.36	82.96
PER	83.77	90.61	87.06	94.35	79.46	86.26
Accuracy	83.81%			82.24%		

On principle our systems perform well with LOC, ORG and PER classes as it can be noticed in Tables 4 and 5, but face difficulties detecting MISC class who can refer to anything from movie titles to sports events, etc.

5 Classifier Combination

It is a well-known fact that if several classifiers are available, they can be combined in various ways to create a system that outperforms the best individual classifier. Since we had several classifiers available, it was reasonable to investigate combining them in different ways. The simplest approach to combining classifiers is through voting, which examines the outputs of the various models and selects the classifications which have a weight exceeding some threshold, where the weight is dependent upon the models that proposed this particular

classification. It is possible to assign various weights to the models, in effect giving one model more importance than the others. In our system, however, we simply assigned to each model equal weight, and selected classifications which were proposed by a majority of models. Voting was thus used to improve further the base model.

In the three separate votings we made a combinations of HMM and the feature set variations of ME and Timbl. In Table 6 voting's results per LOC, MISC and ORG classes are higher in comparison with the one of HMM and Timbl but still lower than ME. PER's score is greater than each individual system and voting's accuracy is only less than the one of ME.

As discussed in Section 4, the reduced set B covers MISC class better than ME, so the second voting was among HMM and the classifiers' reduced set B. In Table 7 for LOC, MISC and PER class voting performs better among the three classifiers and only Timbl has greater coverage with ORG class. Compared to the accuracy of each individual system, the reached 83.95% score is the higher one.

The voting (Table 8) where best performing systems have participated reached 84.15% F-measure for LOC and 84.86% for ORG classes which is higher than the individual performance of HMM for the same classes but less than the results obtained by Timbl and ME. For MISC 71.50% F-measure is achieved, the highest score in comparison not only with a system individually but also with the other votings (Table 6 and 7) we had.

In conclusion applying voting among the best performing systems raised accuracy with 0.11% and led to 71.50% classification of MISC class which is particularly difficult due to its generality as discussed before.

For CoNLL-2002 Carreras[2](Carr.) gained the best score for NE classification. In Table 9 we show our voting results together with the one obtained by their system. It can be seen that we managed to improve classification for each one of the LOC, MISC, ORG and PER classes. Our third voting system im-

Table 6. Voting among Timbl, ME using set A and HMM.

	LOC(%)			MISC(%)			ORG(%)			PER(%)			All
Classif.	Prec.	Rec.	$F_{\beta=1}$	Prec.	Rec.	$F_{\beta=1}$	Prec.	Rec.	$F_{\beta=1}$	Prec.	Rec.	$F_{\beta=1}$	Accur.
Timbl	82.02	80.81	81.41	64.83	61.95	63.35	81.61	84.93	83.23	88.29	85.17	86.70	81.54
ME	88.25	81.09	84.52	85.65	58.11	69.24	80.66	90.29	85.20	86.93	90.48	88.67	84.46
HMM	80.02	73.16	76.43	56.46	59.29	57.84	77.60	77.71	77.66	69.72	76.74	73.06	74.37
Vot 1	86.01	81.64	83.77	82.98	57.52	67.94	79.71	89.21	84.19	89.68	88.71	89.19	83.78

Table 7. Voting among Timbl, ME using set B and HMM.

	LOC(%)			MISC(%)			ORG(%)			PER(%)			All
Classif.	Prec.	Rec.	$F_{\beta=1}$	Prec.	Rec.	$F_{\beta=1}$	Prec.	Rec.	$F_{\beta=1}$	Prec.	Rec.	$F_{\beta=1}$	Accur.
Timbl	85.35	82.20	83.74	77.54	63.13	69.59	83.92	86.5	85.19	83.77	90.61	87.06	83.81
ME	86.33	79.24	82.64	91.19	51.92	66.17	74.64	93.36	82.96	94.35	79.46	86.26	82.24
HMM	80.02	73.16	76.43	56.46	59.29	57.84	77.60	77.71	77.66	69.72	76.74	73.06	74.37
Vot 2	87.00	80.90	83.84	88.0	58.41	70.21	78.71	90.07	84.01	90.04	88.57	89.30	83.95

Table 8. Voting among Timbl using set B, ME using set A and HMM.

	LOC(%)			MISC(%)			ORG(%)			PER(%)			All
Classif.	Prec.	Rec.	$F_{\beta=1}$	Prec.	Rec.	$F_{\beta=1}$	Prec.	Rec.	$F_{\beta=1}$	Prec.	Rec.	$F_{\beta=1}$	Accur.
Timbl	85.35	82.20	83.74	77.54	63.13	69.59	83.92	86.5	85.19	83.77	90.61	87.06	83.81
ME	88.25	81.09	84.52	85.65	58.11	69.24	80.66	90.29	85.20	86.93	90.48	88.67	84.46
HMM	80.02	73.16	76.43	56.46	59.29	57.84	77.60	77.71	77.66	69.72	76.74	73.06	74.37
Vot 3	86.92	81.55	84.15	86.25	61.06	71.50	81.51	88.5	84.86	86.94	92.38	89.58	84.57

Table 9. Comparing voting 1,2,3 with the results of Careras.

	LOC(%)			MISC(%)			ORG(%)			PER(%)			All
Classif.	Prec.	Rec.	$F_{\beta=1}$	Prec.	Rec.	$F_{\beta=1}$	Prec.	Rec.	$F_{\beta=1}$	Prec.	Rec.	$F_{\beta=1}$	Accur.
Vot 1	86.01	81.64	83.77	82.98	57.52	67.94	79.71	89.21	84.19	89.68	88.71	89.19	83.78
Vot 2	87.00	80.90	83.84	88.0	58.41	70.21	78.71	90.07	84.01	90.04	88.57	89.30	83.95
Vot 3	86.92	81.55	84.15	86.25	61.06	71.50	81.51	88.5	84.86	86.94	92.38	89.58	84.57
Carr.	85.76	79.43	82.47	60.19	57.35	58.73	81.21	82.43	81.81	84.71	93.47	88.87	81.39

proved F-measure with 1.68% LOC, 12.77% MISC, 3.05% ORG and 0.71% PER class classification. Each voting observed separately has higher F-measure than the one obtained by Carreras.

6 Conclusions and Future Work

In this paper we present a combination of three different Named Entity Recognition systems based on machine learning approaches applied to Spanish texts. Every named entity system we have introduced is using the same set or subset of features over the same training corpus for tuning the system and the same test corpus for evaluating it. Three different combinations have been developed in order to improve the score of each individual system. The results are encouraging since 98.50% overall accuracy was obtained for NE detection using BIO model and 84.94% for NE classification into LOC, ORG, PER and MISC classes. However, Maximum Entropy as individual system performs better with NE classification into LOC and ORG classes. As a whole we need to improve our scores by adding more information using new features. For future work we also intend to include morphological and semantic information, to develop a more sophisticated voting system and to adapt our system for other languages.

Acknowledgements

This research has been partially funded by the Spanish Government under project CICyT number TIC2000-0664-C02-02 and PROFIT number FIT-340100-2004-14 and by the Valencia Government under project numbers GV04B-276 and GV04B-268.

References

1. Montserrat Arevalo, Montserrat Civit, and Maria Antonia Martí. MICE: A module for Named Entity Recognition and Clasification. *International Journal of Corpus Linguistics*, 9(1):53 – 68, March 2004.
2. Xavier Carreras, Lluís Màrques, and Lluís Padró. Named entity extraction using adaboost. In *Proceedings of CoNLL-2002*, pages 167–170. Taipei, Taiwan, 2002.
3. Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. TiMBL: Tilburg Memory-Based Learner. Technical Report ILK 03-10, Tilburg University, November 2003.
4. Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. Named entity recognition through classifier combination. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 168–171. Edmonton, Canada, 2003.
5. James Mayfield, Paul McNamee, and Christine Piatko. Named entity recognition using hundreds of thousands of features. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 184–187. Edmonton, Canada, 2003.
6. Diana Maynard, Valentin Tablan, Cristian Ursu, Hamish Cunningham, and Yorick Wilks. Named Entity Recognition from Diverse Text Types. In R. Mitkov N. Nicolov G. Angelova, K. Bontcheva and N. Nikolov, editors, *Proceedings of the Recent Advances in Natural Language Processing*, Tzigov Chark, 2001.
7. Adwait Ratnaparkhi. *Maximum Entropy Models For Natural Language Ambiguity Resolution*. PhD thesis, Computer and Information Science Department, University of Pennsylvania, 1998.
8. Tijong Kim Sang. Introduction to the conll-2002 shared task: Language independent named entity recognition. In *Proceedings of CoNLL-2002*, pages 155–158, 2002.
9. Ingo Schröder. A case study in part-of-speech tagging using the icopost toolkit. Technical Report FBI-HH-M-314/02, Department of Computer Science, University of Hamburg, 2002.
10. Armando Suárez and Manuel Palomar. A maximum entropy-based word sense disambiguation system. In Hsin-Hsi Chen and Chin-Yew Lin, editors, *Proceedings of the 19th International Conference on Computational Linguistics, COLING 2002*, pages 960–966, August 2002.