

Greek Named Entity Recognition using Support Vector Machines, Maximum Entropy and Onetime

Ionas Michailidis^{†*}, Konstantinos Diamantaras[†], Spiros Vasileiadis[†], Yannick Frère[‡]

[†] Department of Informatics
Alexander Technological Education Institute of Thessaloniki
57400 Sindos, Greece

* Laboratoire d'Informatique
Ecole Polytechnique de l'Université "François Rabelais" de Tours
64 avenue Jean Portalis,
37200 Tours, France

‡ Haute École de la Province de Liège "Rennequin Sualem"
Quai Gloesener, 6
4020 Liège, Belgium

ionas@teithe.gr, kdiamant@it.teithe.gr, spyridon.vasileiadis@googlemail.com, yannick_frere@hotmail.com

Abstract

We describe our work on Greek Named Entity Recognition using comparatively three different machine learning techniques: (i) Support Vector Machines (SVM), (ii) Maximum Entropy and (iii) Onetime, a shortcut method based on previous work of one of the authors. The majority of our system's features use linguistic knowledge provided by: morphology, punctuation, position of the lexical units within a sentence and within a text, electronic dictionaries, and the outputs of external tools (a tokenizer, a sentence splitter, and a Hellenic version of Brill's Part of Speech Tagger). After testing we observed that the application of a few simple Post Testing Classification Correction (PTCC) rules created after the observation of output errors, improved the results of the SVM and the Maximum Entropy systems output. We achieved very good results with the three methods. Our best configurations (Support Vector Machines with a second degree polynomial kernel and Maximum Entropy) achieved both after the application of PTCC rules an overall F-measure of 91.06.

1. Introduction

Named Entity Recognition (NER) can be described as the task of recognizing and classifying single and multi-word expressions within a document that act as proper names of any kind (person names, location names, organization names, etc.). For example, in the text: "*Ο πρόεδρος της Ευρωπαϊκής Επιτροπής κ. Ζακ Σαντέρ συναντήθηκε στις Βρυξέλλες με τον πρωθυπουργό της Μ. Βρετανίας κ. Μπλερ.*" a NER system should recognize "*Ζακ Σαντέρ*" (*Jacques Santer*) and "*Μπλερ*" (*Blair*) as Person Named Entities, "*Βρυξέλλες*" (*Brussels*) and "*Μ. Βρετανίας*" (*Great Britain*) as Location Named Entities, and "*Ευρωπαϊκής Επιτροπής*" (*European Commission*) as an Organization Named Entity.

NER systems are essential components of Information Extraction and Question Answering systems. NER is also used for text indexing and classification (Friburger & Maurel, 2002), as well as in parallel corpora alignment (Steinberger et al., 2004).

The benchmarks of the NER domain are considered to be those established by four major international conferences that took place in the form of competitions of NER systems: MUC-6¹ (Chinchor, 1995) (in 1995), MUC-7² (Chinchor, 1998) (in 1997) CoNLL-2002 shared task³ (Tjong Kim Sang, 2002) and CoNLL-2003 shared task⁴ (Tjong Kim Sang & De Meulder, 2003). During the

first two conferences (MUC) the participating systems where asked to recognize and classify person, location, and organization proper names as well as date, time, percentage, and monetary value expressions. The target language was English. In the latest two conferences (CoNLL shared task conferences) the participating systems had to recognize and classify person, location, and organization entities, as well as miscellaneous entities that do not belong to the three previous groups like for example names of artifacts, book and film titles etc. The target languages where: Dutch and Spanish for the 2002 edition, and English and German for the 2003 edition. The different teams had to participate in both language evaluations, since the goal of the conferences was the creation of language independent systems.

In the past, various methods have been used by NER systems. Depending on the method, we can classify these systems into three general categories:

- Systems using hand-crafted recognition grammars, like the New York University's system (Grishman, 1995) involved in MUC-6;
- Systems using "machine learning" techniques, like all the systems used in the CoNLL shared task conferences. Some of them did very well on both languages;
- Hybrid systems using both approaches, for example, the very successful system of the Language Technology Group at the University of Edinburgh (Mikheev et al., 1998) presented in MUC-7.

In previous work on Greek NER Karkaletsis et al. (1999) use a hand-crafted grammar system and a machine learning algorithm (C4.5) to automatically acquire NER grammars when moving to a new domain. Farmakiotou et al. (2000) use a system based on hand-crafted lexical

¹ <http://www.cs.nyu.edu/cs/faculty/grishman/muc6.html>

² http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/muc_7_toc.html

³ <http://www.cnts.ua.ac.be/conll2002/ner/>

⁴ <http://www.cnts.ua.ac.be/conll2003/ner/>

ressources. Petasis et al. (2002) use the C4.5 machine learning algorithm to update NER grammars. Boutsis et al. (2000) use a collection of 110 hand-crafted grammars. Lucarelli (2005) uses Support Vector Machines to recognize person Named Entities and semi-automatically created patterns to recognize temporal expressions.

Our experiment concerns “CoNLL-2002 shared task”-style, Greek NER using comparatively three machine learning approaches: (i) Support Vector Machines (Vapnik, 1995), (ii) Maximum Entropy (Berger et al., 1996) and (iii) Onetime (Diamantaras et al. 2005). In the following sections we will make a description of the corpora we used (section 2) and the features we extracted for every token of the corpora (section 3). Follows, a discussion on how to approach NER as a machine learning problem (section 4) and a short presentation of the three machine learning techniques we used (sections 5, 6 and 7). Finally we present our experiment (section 8), a method for improving our system’s output (section 9), the experimental results and a discussion on these results (section 10).

2. The corpus

The Greek corpus we have used comes from the newspaper “TA NEA”. It is a selection of 400 articles from the year 1997 covering diverse columns of the newspaper («Γνώμες – Σχόλια» (*Opinions*), «Πολιτική» and «Μικροπολιτικός» (*Politics News*), «Κοινωνία» (*Society*), «Κόσμος» (*International News*), «Οικονομία» (*Economic News*), «Πανόραμα» (*Art News*), «Ομάδα» (*Sport News*), «Επιστολές» (*Letters from the readers*), «Θεάματα» (*Spectacles*). The corpus measures approximately 172,000 tokens and is divided into 400 different files (400 articles). According to Ellogon Components Specifications (Afantenos et al., 2002) we consider as tokens all character sequences in a text which are separated by blank spaces or punctuation marks. Additionally punctuation marks are also considered to be tokens. Similarly to Boutsis et al. (2000) the selected articles contain a high percentage of tokens beginning with a capital letter.

We have divided the corpus into two parts: the first part (approximately 138,000 tokens) is used for training our system, while the second (approximately 34,000 tokens) is used for testing it. The criterion we used for dividing the corpus is the author of the article. The authors of the articles of the training corpus are different from the ones of the test corpus. We did this in order to evaluate our system with articles that may have a different writing style from the articles of the training corpus and thus increase the degree of difficulty of the task. We then used the tools provided with the Ellogon Text Engineering Platform (Petasis et al., 2002b) in order to:

- split the two corpora into sentences and tokens (use of a Sentence Splitter and a Tokenizer) without manual correction
- obtain Part of Speech (PoS) tags for all tokens of the two corpora (use of the Hellenic version of Brill’s PoS Tagger (Brill, 1992; Afantenos et al., 2002)) without manual correction
- manually tag the Named Entities in the two corpora following CoNLL-2002 annotation guidelines (use of a corpus annotation tool)

- create the input of the feature extraction tool for both corpora (one file where each line contains: the token, its Named Entity tag according to CoNLL-2002 shared task guidelines, a binary information indicating if the current token is the first token of a sentence, its PoS tag as it is provided by Brill’s PoS Tagger, and the name of the article the token belongs to)

According to the CoNLL-2002 shared task annotation guidelines every Named Entity token is associated with a Named Entity tag encoding its Named Entity category (Person, Location, Organization, Miscellaneous) and its position in the Named Entity (whether it is the first token of a Named Entity or not). Tokens tagged with O are Outside of Named Entities. The B-X tag is used for the token at the Beginning of a Named Entity of type X while the I-X tag is used for all other tokens Inside the Named Entity of type X. The X Entities are of four types: persons (PER), locations (LOC), organizations (ORG), and miscellaneous (MISC). This tagging scheme is a variant of the IOB scheme originally proposed by Ramshaw and Marcus (1995). An example of the input of the feature extraction tool is given on figure 1.

TOKEN	NE TAG	FIRST TOKEN OF A SENTENCE	PoS TAG	FILE-NAME
Ο	O	YES	DDT	File1.txt
κ.	O	NO	NNF	File1.txt
Ζακ	B-PER	NO	NNPM	File1.txt
Σαντέρ	I-PER	NO	NNPM	File1.txt
συναντήθηκε	O	NO	VBD	File1.txt
στις	O	NO	DDT	File1.txt
Βρυξέλλες	B-LOC	NO	NNPSF	File1.txt
με	O	NO	IN	File1.txt
τον	O	NO	DDT	File1.txt
κ.	O	NO	NNF	File1.txt
Μπλερ	B-PER	NO	NNPM	File1.txt
.	O	NO	.	File1.txt

Figure 1. Input of the feature extraction tool

The PoS tags are the ones attributed by the Hellenic Version of the Brill’s PoS tagger. They are further described in (Afantenos et al. 2002). During the annotation process Named Entities were assumed to be non-recursive and non-overlapping. In case of a Named Entity embedded in another Named Entity only the top level Entity was marked.

3. The Features

The features we used for both training and evaluating the system were mainly inspired by Chieu and Ng (2003). The features, which are all binary (they respond to questions about a token with “Yes” → ”1” or “No” → ”0”), result in an array of zeros and ones. The array size is M×N where M is the number of tokens of the corpus and N is the total number of features.

There are two types of features: local and global features. Local features make use of information related to the current token or its close context. Global features make use of information related to the whole article. The

total number of features in our system is 91 (89 local and 2 global).

3.1. Local features

The local features can be grouped into features that concern the current token, the previous, the previous two, the next, and the next two tokens. Local features use information relating to morphology, punctuation, PoS, dictionary membership and list membership. Below we describe the features according to these groups.

3.1.1. Morphological features

The morphological features relate to morphological observations about the current token. They correspond to the following binary tests:

- Token begins with a capital letter (Yes / No)
- All letters of the token are capital (Yes / No)
- Token is composed of only one capital letter (Yes/No)
- Token contains at least one digit (Yes / No)
- Token has only Latin characters (Yes / No)

3.1.2. Punctuation features

The punctuation features relate to observations about the current token in relation with punctuation and the position of the token in the sentence and the article. They correspond to the following binary tests:

- Token is the first token of a sentence (Yes / No)
- Token is the first token of an article (Yes / No)
- Token follows an opening parenthesis (Yes / No)
- Token precedes a closing parenthesis (Yes / No)
- Token follows an opening («) or an ASCII (") quote (Yes/No) (2 different features)
- Token precedes a closing (») or an ASCII (") quote (Yes / No) (2 different features)

3.1.3. PoS features

The PoS features are related to observations about the current, the previous and the next token's PoS tag attributed by the Hellenic version of the Brill's PoS tagger. We use a reduced tag set (16 different tags) of the different PoS tags attributed by the Hellenic version of the Brill's PoS tagger. The features correspond to the following binary tests:

- Token has a V⁵ PoS tag (Yes / No) (16 different features)
- Token follows a token with a V PoS tag (Yes / No) (16 different features)
- Token precedes a token with a V PoS tag (Yes / No) (16 different features)

3.1.4. Dictionary features

The dictionary features concern the current and the previous token in relation to their membership in electronic dictionaries containing all the inflected forms of some word classes. The features correspond to the following binary tests:

- Token is a part of a dictionary of person first names (Yes / No) (Mavropoulos & Bakoura, 2002)
- Token is a part of a dictionary of location names (Yes / No) (Bakoura, 2003)
- Token follows a token which is part of a dictionary of profession names (Yes / No) (Foufi, 2004)

⁵ V = each 1 of the 16 possible PoS tags

- Token ends with a family name suffix contained in a dictionary of family name suffixes (Yes / No)

3.1.5. List features

The list features are concerned with the current and the previous two tokens in relation to some lists of tokens derived from the training corpus. The features correspond to the following binary tests:

- Token is a part of a list containing non-capitalized tokens found at least 2 times in the training corpus tagged with the Y⁶ Named Entity tag (Yes / No) (8 different features). For example in the text «*Τράπεζα της Ελλάδος*» (*Bank of Greece*), «*της*» (*of*) is a non-capitalized token tagged as I-ORG.
- Token is a part of a list containing tokens found at least 2 times in the training corpus tagged with the Y Named Entity tag (Yes / No) (8 different features)
- Previous token is a part of a list containing tokens in the training corpus preceding at least 2 times a Named Entity token of the Z⁷ category (Yes / No) (4 different features). For example in the text «*στην Ιταλία*» (*in Italy*), «*στην*» (*in*) is a token preceding a LOC Named Entity.
- Previous 2 tokens are a part of a list containing sequences of 2 consecutive tokens in the training corpus preceding at least 2 times a Named Entity token of the Z category (Yes / No) (4 different features). For example in the text «*με τον Κώστα Βουτσά*» (*with Kostas Voutsas*), «*με τον*» (*with*) is a sequence of two tokens preceding a PER Named Entity.

3.2. Global features

The global features are the two following:

- We search the capitalized first token of a sentence elsewhere in the same article. If this token is found, is not the first token of a sentence, and is capitalized, this feature is set to "Yes" → "1". This feature helps identifying tokens that begin with a capital letter independently of their position in the sentence.
- If the token is made up of all capitalized letters we look in the same article for sequences of tokens whose capitalized initials match the token in question. If found, this feature is set to "Yes" → "1". This feature helps identifying acronyms.

4. NER and machine learning

The NER problem using the CoNLL-2002 shared task scheme is the automatic assignment of a Named Entity tag to every token of a text. This problem can be seen as a classification task where every token of the text must be classified in one of the nine Named Entity tag classes (B-PER, I-PER, B-LOC, I-LOC, B-ORG, I-ORG, B-MISC, I-MISC and O for tokens not belonging to Named Entities). In order to solve a classification problem we use a classification algorithm. Classification algorithms can be grouped, depending on the number of classes they can handle, into multi-class like Maximum Entropy and binary like Support Vector Machine and Onetime. Our NER

⁶ Y = each one of the possible Named Entity tags, i.e. B-PER, I-PER, B-LOC, I-LOC, B-ORG, I-ORG, B-MISC, I-MISC (excluding the non Named Entity tag "O")

⁷ Z = each one of the four Named Entity categories, i.e. PER, LOC, ORG, MISC

problem involves nine classes while SVM and Onetime are binary classification algorithms. How to solve a multi-class problem using a binary classification algorithm?

We can face this issue by transforming the single nine class problem into several binary class problems (solved by several classifiers) and using an appropriate strategy to combine the outputs of all the classifiers into a unique decision for every token. Our previous work (Michailidis et al. 2005) showed that the transformation of a Greek NER multi-class problem into several binary class ones is done better using the “one versus one” (Fuernkranz, 2002) method (also called “Round Robin” or “pair-wise”).

The “one versus one” method uses a binary classifier for each combination pair of the given output classes of the training set. If we have i output classes we need $i*(i-1)/2$ binary classifiers. In our case $i=9$ so there are 36 classifiers. For each combined pair, only the samples which represent the two current output classes are considered during the training process. In order to combine the decisions from the multiple classifiers we applied simple voting.

5. Support Vector Machines

The Support Vector Machines (SVM) is a binary classification technique (Vapnik, 1995). SVM faces the problem of the two class separation of the input space of M N -dimensional samples (training set) using an N -dimensional hyperplane, by mapping with a kernel function the input space into a high-dimensional space F , where the optimal separation hyperplane can be easily calculated. The SVM performs classification on the high dimensional space F by choosing the optimal hyperplane which maximizes the margin of any point in the training set. Statistical learning theory suggests that, for well-behaved classes, the choice of the maximum margin hyperplane will lead to maximal generalization when predicting the class membership of previously unseen samples. For more detailed discussion about this algorithm please refer to Vapnik (1995) and Burges (1998). Our system uses the Support Vector Machines implementation “SVM light”⁸ written by Joachims (1999). We trained and tested our SVM system using three different kernel functions. Those were: (i) the Linear, (ii) the Polynomial and (iii) the Gaussian or Radial Basis Function (RBF) kernel.

6. Maximum Entropy

According to Chieu and Ng (2004), the Maximum Entropy method estimates probabilities based on the principle of making as few assumptions as possible, other than the constraints imposed. Such constraints are derived by training data, expressing some relationship between features and outcome. The probability distribution which satisfies the above property is the one with the highest entropy. It is unique, agrees with the maximum-likelihood distribution, and has the exponential form:

$$p(o|h) = \frac{1}{Z(h)} \prod_{j=1}^k \alpha_j^{f_j(h,o)}$$

where o refers to the outcome, h the history (or context), and $Z(h)$ is a normalization function to ensure that

$\sum_o p(o|h)=1$. In addition, each feature function $f_j(h,o)$ is a binary function. An example of a feature function is:

$$f_j(h,o) = \begin{cases} 1 & \text{if } o=B\text{-PER, token starts with capital letter} \\ 0 & \text{otherwise} \end{cases}$$

The parameters α_j can be estimated by a procedure called Generalized Iterative Scaling (GIS) (Darroch & Ratcliff, 1972). The Maximum Entropy method is further described by Berger et al. (1996). In our experiment we used the Maximum Entropy toolkit named “MaxEnt”⁹ written by Zangh Le (2004).

7. Onetime

The Onetime algorithm (Diamantaras et al., 2005) is a shortcut method for binary linear classification based on previous work of one of the authors (Diamantaras & Strintzis 1998).

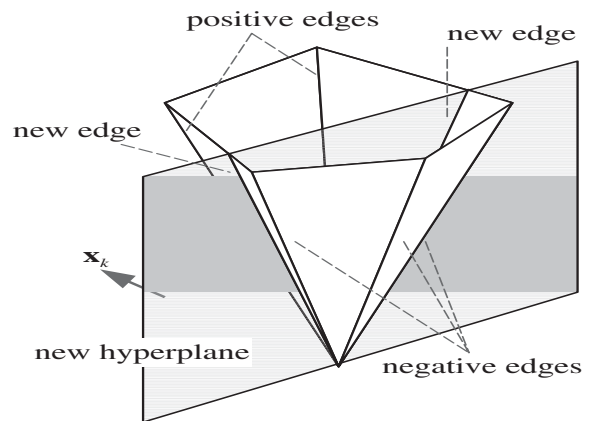


Figure 2. The solution cone

This earlier work focused on the description of the solution cone (see fig. 2) through a set of edges and it proposed a recursive algorithm for updating those edges. Onetime is an extension of that idea based on the fact that we do not need the exact description of the solution cone, but only one solution vector inside it. The learning algorithm leaves intact the solution vector when it is inside the new solution cone defined by the positive part of the old solution cone and the new hyperplane (corresponding to the new pattern x_k). The solution vector is updated when it is not inside the new solution cone. The update is done by finding a new solution vector which is “barely” inside the new solution cone.

8. Experiment

Our training corpus featured ca 138,000 data samples. The test corpus counted ca 34,000 test samples. The number of features was 91. Our experiment used comparatively the above corpora for training and testing three systems based on the machine learning approaches of: (i) Support Vector Machines, (ii) Maximum Entropy and (iii) Onetime.

9. Post Testing Classification Correction

Studying the output results of our tests, we observed several repeating misclassification patterns. More

⁸ <http://svmlight.joachims.org/>

⁹ http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

specifically, the SVM and the Maximum Entropy systems failed to assign the correct Named Entity tags sequences in cases like the ones of the SVM case shown in table 1. In the right column we can see the correct sequences of Named Entity tags and on the left the wrong ones followed by the number of times each sequence appears in the output. We developed simple heuristic Post Testing Classification Correction (PTCC) rules which detect and correct errors in the system's output.

Example rule:

B-MISC O I-MISC → B-MISC I-MISC I-MISC

The results of PTCC were remarkable on the affected Named Entities (ORG and MISC). As can be seen in table 2 the recall for MISC on the top performing configuration of our SVM system climbed from 61% to 77%.

SVM's output sequence of Named Entity Tags	Correct sequence of Named Entity Tags
B-MISC O I-MISC (41 times)	B-MISC I-MISC I-MISC
B-MISC O O I-MISC (30 times)	B-MISC I-MISC I-MISC I-MISC
B-MISC O O O I-MISC (15 times)	B-MISC I-MISC I-MISC I-MISC I-MISC
B-MISC I-MISC O I-MISC (8 times)	B-MISC I-MISC I-MISC I-MISC
B-PER O I-PER (15 times)	B-PER I-PER I-PER
B-PER I-PER O I-PER (13 times)	B-PER I-PER I-PER I-PER
B-ORG O I-ORG (7 times)	B-ORG I-ORG I-ORG
B-ORG I-ORG O I-ORG (5 times)	B-ORG I-ORG I-ORG I-ORG
B-ORG I-ORG O O I-ORG (3 times)	B-ORG I-ORG I-ORG I-ORG I-ORG

Table 1. Post Testing Classification Correction Rules

10. Results and discussion

Named Entity Recognition systems are evaluated using precision, recall and F-measure scores. The precision of a system is the percentage of correctly attributed Named Entity tags among all the Named Entity tags attributed by the system. The recall of a system is the percentage of correctly attributed Named Entity tags compared to a possible perfect NER output. The F-measure is a combination of precision and recall calculated by the following formula:

$$F = \frac{(\beta^2 + 1) * precision * recall}{(\beta^2 * precision + recall)}$$

The choice of β gives more importance to either precision or recall. Table 2 shows all the experimental results for each Named Entity category (PER, LOC, ORG, and MISC) as well as for all the Named Entity categories (OVERALL). We can see that all the methods that do not use the PTCC rules (except the SVM with the RBF kernel) have similar results (overall $F_{\beta=1}$ measure around 87%). The SVM with the RBF kernel has high precision but poor recall, ie. it classifies correctly, but skips too much data. The application of the PTCC rules improved the results in all the experiments except in the case of the Onetime method.

The top performance configurations on the experiment were the SVM with a second-degree Polynomial kernel

function, and the Maximum Entropy, both applying PTCC rules. They scored an overall $F_{\beta=1}$ measure of 91.06 % with very close values of precision and recall.

Further work includes the study of the features playing the most important role in the recognition process, the fine-tuning of the Onetime method, the application of the Viterbi (1967) algorithm to the output of the Maximum Entropy system in order to eliminate sequences of Named Entity tags which are impossible to occur (for example B-PER followed by I-LOC), the extension of the corpus with a Named Entity annotated corpus of 130,000 tokens coming from the newspaper "KATHIMERINI", and the combination of the above described methods with hand-crafted finite state grammars in order to produce a hybrid NER system.

11. References

- Afantenos, S., Petasis, G. and Karkaletsis, V. (2002). "Ellogon components Specifications". National Center of Scientific Research "Demokritos", Institute of Informatics and Telecommunications, Software and Knowledge Engineering Laboratory. <http://www.ellogon.org>.
- Bakoura, S. (2003). "Morphological Description of Anthroponyms and Toponyms in modern Greek". *22ème Colloque International Grammaires et Lexiques compares*. Thessaloniki, Greece.
- Berger, A.L., Della Pietra, S.A., Della Pietra, V.J. (1996). "A Maximum Entropy Approach to Natural Language Processing". *Computational Linguistics*, v.22 n.1. p.39-71.
- Boutsis, S., Demiros, I., Giouli, V., Liakata, M., Papageorgiou, H. and Piperidis, S. (2000). "A system for Recognition of Named Entities in Greek". *Lecture Notes in Computer Science*.
- Brill, E. (1992). "A simple rule-based part of speech tagger". In *Proceedings of the Third Conference on Applied Natural Language Processing, ACL*. Trento, Italy.
- Burges, C.J.C. (1998). "A tutorial on Support Vector Machines for Pattern Recognition". *Data Mining and Knowledge Discovery*. number 2, vol. 2. pp. 121-167.
- Chieu, H.L. and Ng, H.T. (2003). "Named Entity Recognition with a Maximum Entropy Approach", in *Proceedings of CoNLL-2003*. Edmonton, Canada.
- Chinchor, N. (1995). "MUC-6 Named Entity Task Definition (version 2.1)". In *Proceedings of the 6th Message Understanding Conference (MUC-6)*. Columbia, Maryland. Morgan-Kaufmann.
- Chinchor, N. (1998). "MUC-7 Named Entity Task Definition (version 3.5)". In *Proceedings of the 7th Message Understanding Conference (MUC-7)*. Fairfax, VA.
- Darroch, J.N. and Ratcliff, D. (1972). Generalized Iterative Scaling for Log-Linear Models. *Annals of Mathematical Statistics*.43(5). pp.147-1480.
- Diamantaras, K., Michailidis, I. and Vasileiadis, S. (2005). "A very fast and efficient linear classification algorithm". In *Proceedings of the 2005 IEEE Int. Workshop on Machine Learning for Signal Processing (MLSP-2005)*. Mystic, CT, USA.
- Diamantaras, K. and Strintzis, M.G. (1998). "Neural Classifiers using One-Time updating". *IEEE*

- Transactions on Neural Networks*. vol. 9, no 3. pp 436-447.
- Farmakiotou, D., Karkaletsis, V., Koutsias, J., Sigletos, G., Spyropoulos, C.D., and Stamatopoulos, P. (2000). "Rule-based Named Entity recognition for Greek Financial Texts". In *Proceedings of the Workshop on Computational lexicography and Multimedia Dictionaries (COMPLEX 2000)*. pp. 75-78.
- Foufi, V. (2004). "Electronic Dictionary of Greek profession names", in *Proceedings of the 25th annual meeting of the Linguistics Section of the Department of Philology at the Aristotle University of Thessaloniki*. Thessaloniki, Greece.
- Friburger, N. and Maurel, D. (2002). "Textual similarity based on Proper Names". In *Proceedings of the workshop Mathematical / Formal methods in Information Retrieval (MFIR '2002) at the 25th ACM SIGIR Conference*. Tampere, Finland. pp. 155-167.
- Fuernkranz, J. (2002). "Round Robin Classification", *The Journal of Machine Learning Research*. vol.2. pp. 721-747.
- Grishman, R. (1995). "The NYU system for MUC-6 or where is the syntax?". In *Proceedings of the 6th Message Understanding Conference (MUC-6)*. Columbia, Maryland. Morgan-Kaufmann,
- Joachims, T. (1999). "Making large-Scale SVM Learning Practical". *Advances in Kernel Methods - Support Vector Learning*. Schölkopf, B., Burges, C. and Smola, A. (ed.), MIT Press. Cambridge, USA.
- Karkaletsis, V., Paliouras, G., Petasis, G., Manousopoulou, N. and Spyropoulos, C.D. (1999). "Named-Entity Recognition from Greek and English Texts". *Journal of Intelligent and Robotic Systems*. pp. 123-135.
- Lucarelli, G. (2005). "Αναγνώριση και κατάταξη ονομάτων οντοτήτων σε ελληνικά κείμενα". Διπλωματική εργασία Προγράμματος Μεταπτυχιακών Σπουδών στην Επιστήμη των Υπολογιστών, Οικονομικό Πανεπιστήμιο Αθηνών.
- Μανρωπούλου, Α. and Βακούρα, Σ. (2002). "Ηλεκτρονικό λεξικό της νέας ελληνικής: Αναπαράσταση των κυρίων ονομάτων και ουσιαστικών και επιθέτων που δηλώνουν εθνικότητα.". In *Proceedings of the 23rd annual meeting of the Linguistics Section of the Department of Philology at the Aristotle University of Thessaloniki*. Thessaloniki, Greece.
- Michailidis, I., Diamantaras, K. and Vasileiadis, S. (2005). "Greek Named Entity Recognition using Support Vector Machines". *7th International Conference on Greek Linguistics (ICGL-7)*. York, UK.
- Mikheev, A., Grover, C. and Moens, M. (1998). "Description of the LTG System Used for MUC-7". In *Proceedings of the 7th Message Understanding Conference (MUC-7)*. Fairfax, VA.
- Petasis, G., Karkaletsis, V., Paliouras, G., Androutsopoulos, I. and Spyropoulos, C.D. (2002b). "Ellogon: A New Text Engineering Platform". In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*. Las Palmas, Canary Islands, Spain. pp. 72-78.
- Petasis, G., Vichot, F., Wolinski, F., Paliouras, G., Karkaletsis, V. and Spyropoulos, C. (2002). "Using Machine Learning to maintain rule-based Named-Entity Recognition and Classification Systems". *Meeting of the Association for Computational Linguistics (ACL)*.
- Ramshaw, L.A., and Marcus, M.P. (1995). "Text chunking using transformation-based learning". In *Proceedings of the Third ACL Workshop on Very Large Corpora*. Cambridge, MA, USA. pp. 82-94.
- Steinberger, R., Pouliquen, B. and Ignat, C. (2004). "Exploiting multilingual nomenclatures and language-independent text features as an interlingua for cross-lingual text analysis applications". In *Proceedings of the 4th Slovenian Language Technology Conference. Information Society 2004 (IS'2004)*. Ljubljana, Slovenia.
- Tjong Kim Sang, E.F. (2002). "Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition". In *Proceedings of CoNLL-2002*. Taipei, Taiwan. pp. 155-158.
- Tjong Kim Sang, E.F. and De Meulder, F. (2003). "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition". In *Proceedings of CoNLL-2003*. Edmonton, Canada. pp. 142-147.
- Vapnik, V.N. (1995). *The Nature of Statistical Learning Theory*, Springer-Verlag.
- Viterbi, A.J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory* 13(2). pp. 260-267.
- Zangh Le. (2004). Maximum Entropy Modeling Toolkit for Python and C++.

METHOD	PERSON			LOCATION			ORGANIZATION			MISCELLANEOUS			OVERALL		
	Precision	Recall	F _{p-1}	Precision	Recall	F _{p-1}	Precision	Recall	F _{p-1}	Precision	Recall	F _{p-1}	Precision	Recall	F _{p-1}
SVM linear	95.7	96.18	95.94	88.71	88.58	88.64	83.84	74.29	78.78	88.9	59.44	71.25	91.39	83.75	87.4
SVM linear + PTCC	95.7	97.15	96.42	88.96	88.58	88.77	83.9	75.53	79.49	91.02	76.26	82.99	91.71	87.62	89.61
SVM polynomial	96.46	96.61	96.53	89.56	89.3	89.43	85.76	79.32	82.42	90.57	60.83	72.78	92.46	85.36	88.77
SVM polynomial + PTCC	96.46	97.7	97.07	90.08	89.3	89.69	85.95	81.62	83.73	92.3	77.15	84.05	92.78	89.41	91.06
SVM RBF	98.46	72.28	83.36	94.24	42.63	58.70	88.72	53.53	66.77	92.54	39.26	55.13	95.28	58.38	72.40
SVM RBF + PTCC	98.41	72.55	83.53	94.24	42.63	58.70	88.85	54.94	67.90	93.29	45.40	61.07	95.32	59.95	73.61
Maximum Entropy	96.57	95.25	95.90	90.04	87.57	88.79	83.93	76.15	79.85	88.80	55.69	68.45	91.99	82.86	87.19
Maximum Entropy + PTCC	96.86	97.50	97.18	90.08	89.30	89.69	85.76	81.97	83.83	92.25	76.55	83.67	92.91	89.28	91.06
OneTime	95.31	94.90	95.10	88.18	88.44	88.31	85.05	76.41	80.50	89.40	57.57	70.04	91.25	83.22	87.05

Table 2. Experimental results