# Named Entity Recognition Using a New Fuzzy Support Vector Machine

**Alireza Mansouri, Lilly Suriani Affendey, Ali Mamat**

Faculty of Computer Science & Information Technology, University Putra Malaysia, 43400 Serdong, Malaysia

**Summary**
Recognizing and extracting exact name entities, like Persons, Locations, Organizations, Dates and Times are very useful to mining information from electronics resources and text. Learning to extract these types of data is called Named Entity Recognition (NER) task. Proper named entity recognition and extraction is important to solve most problems in hot research area such as Question Answering and Summarization Systems, Information Retrieval and Information Extraction, Machine Translation, Video Annotation, Semantic Web Search and Bioinformatics, especially Gene identification, proteins and DNAs names. Nowadays more researchers use three type of approaches namely, Rule-base NER, Machine Learning-base NER and Hybrid NER to identify names. Machine learning method is more famous and applicable than others, because it's more portable and domain independent. Some of the Machine learning algorithms used in NER methods are, support vector machine (SVM), Hidden Markov Model, Maximum Entropy Model (MEM) and Decision Tree. In this paper, we review these methods and compare them based on precision in recognition and also portability using the Message Understanding Conference (MUC) named entity definition and its standard data set to find their strength and weakness of each these methods. We have improved the precision in NER from text using the new proposed method that calls FSVM for NER. In our method we have employed Support Vector Machine as one of the best machine learning algorithm for classification and we contribute a new fuzzy membership function thus removing the Support Vector Machine's weakness points in NER precision and multi classification. The design of our method is a kind of One-Against-All multi classification technique to solve the traditional binary classifier in SVM.

***Key words:***
*Named Entity Recognition and Extraction, Information Retrieval, Information Extraction, Text retrieval, Feature Selection, Video Annotation*

## 1. Introduction

Named Entity Recognition (NER) is a subproblem of information extraction and involves processing structured and unstructured documents and identifying expressions that refer to peoples, places, organizations and companies. NER is a fundamental task and it is the core of natural language processing (NLP) system. NER involves two tasks, which is firstly the identification of proper names in text, and secondly the classification of these names into a set of predefined categories of interest, such as person names, organizations (companies, government organisations, committees, etc), locations (cities, countries, rivers, etc), date and time expressions. The term Named Entity was introduced in the sixth Message Understanding Conference (MUC-6). In fact, the MUC conferences were the events that have contributed in a decisive way to the research of this area. It has provided the benchmark for named entity systems that performed a variety of information extraction tasks [1].

For humans, NER is intuitively simple, because many named entities are proper names and most of them have initial capital letters and can easily be recognized by that way, but for machine, it is so hard. One might think the named entities can be classified easily using dictionaries, because most of named entities are proper nouns, but this is a wrong opinion. As time passes, new proper nouns are created continuously.

Therefore, it is impossible to add all those proper nouns to a dictionary. Even though named entities are registered in the dictionary, it is not easy to decide their senses. Most problems in NER are that they have semantic (sense) ambiguity; on the other hand, a proper noun has Different senses according to the context [12]. For illustration, when is "*The White house*" an organization, and when is it a location? When is "*June*" a person name? And when is it a month name? Or in "He visited Bush at White House", here White House is a location", but in "White House announced the list of ministry candidate", White House is an organization.

Automatically extracting proper names is useful to many problems such as machine translation, information retrieval, question answering and summarization. For instance, the key to a question processor is to identify the asking point (who, what, when, where, etc), so in many cases the asking point corresponds to a NE. In biology text data, the named entity system, can automatically extract the predefined names (like protein and DNA names) from raw documents. The goal of named entity recognition and extraction is to extract and classify names into some particular categories from text by respect to the sense of names. The rest of this paper is organized as follows. In Section 2, we review previous related works and investigate three types of existing methods. Section 3

Manuscript received February 5, 2008
Manuscript revised February 20, 2008

introduces the Message Understanding Conference (MUC) definitions, scopes and evaluation parameters for NER and we compare existing methods base on this evaluation metrics. In Section 4 we propose a new fuzzy NER system. In section 5 we draw the conclusion and future work.

## 2. Related Works

In recent years, automatic named entity recognition and extraction systems have become one of the popular research area that a considerable number of studies have been addressed on developing these systems. They can be categorized into three classes [2], namely, Hand-made Rule-based NER, Machine Learning-based NER and Hybrid NER.

Hand made Rule-based approaches focuses on extracting names using lots of human-made rules sets. Generally the systems consist of a set of patterns using grammatical (e.g. part of speech), syntactic (e.g. word precedence) and orthographic features (e.g. capitalization) in combination with dictionaries [3]. An example for this type of system is: "President Bush said Monday's talks will include discussions on security, a timetable for U.S. forces to leave Iraq". In this example a proper noun follows a person's title(president), then noun is a person's name and proper noun that is started with capital character (Iraq) after the verb (to leave) is a Location's name. In this family of approaches, Appelt et. al. [13,17], propose a name identification system based on carefully handcrafted regular expression called FASTUS. They divided the task into three steps: Recognizing Phrases, Recognizing Patterns and Merging incidents, while Iwanska [14] uses extensive specialized resources such as gazetteers, and white and yellow pages. Morgan, for the same purpose, uses a highly sophisticated linguistic analysis [15], Grishman introduce NYU systems that use handcrafted rules[16]. These approaches are relying on manually coded rules and manually compiled corpora. These kinds of models have better results for restricted domains, are capable of detecting complex entities that learning models have difficulty with. However, the rule-based NE systems lack the ability of portability and robustness, and furthermore the high cost of the rule maintains increases even though the data is slightly changed. These type of approaches are often domain and language specific and do not necessarily adapt well to new domains and languages.

In Machine Learning-based NER system, the purpose of Named Entity Recognition approach is converting identification problem into a classification problem and employs a classification statistical model to solve it. In this type of approach, the systems look for patterns and relationships into text to make a model using statistical models and machine learning algorithms. The systems identify and classify nouns into particular classes such as persons, locations, times, etc base on this model, using machine learning algorithms.

There are two types of machine learning model that are use for NER. Supervised and Unsupervised machine learning model. Supervised learning involves using a program that can learn to classify a given set of labeled examples that are made up of the same number of features. Each example is thus represented with respect to the different feature spaces. The learning process is called supervised, because the people who marked up the training examples are teaching the program the right distinctions.

The supervised learning approach requires preparing labeled training data to construct a statistical model, but it cannot achieve a good performance without a large amount of training data, because of data sparseness problem. In recent years several statistical methods based on supervised learning method were proposed. Bikel et. al. propose a learning name-finder base on hidden Markov model [8] called Nymbel, while Borthwick et. al. investigates exploiting diverse knowledge sources via maximum entropy in named entity recognition [9,10]. A tagging of unknown proper names system with Decision Tree model was proposed by Bechet et. al. [5], while Wu et. al. presented a named entity recognition system based on support vector machines [2].

Unsupervised learning method is another type of machine learning model, where an unsupervised model learns without any feedback. In unsupervised learning, the goal of the program is to build representations from data. These representations can then be used for data compression, classifying, decision making, and other purposes. Unsupervised learning is not a very popular approach for NER and the systems that do use unsupervised learning are usually not completely unsupervised. In these types of approach, Collins et. al. Discusses an unsupervised model for named entity classification by use of unlabeled examples of data [7] , Koim et. al. Proposes an unsupervised named entity classification models and their ensembles that uses a small-scale named entity dictionary and an unlabeled corpus for classifying named entities [4]. Unlike the rule-based method, these types of approaches can be easily port to different domain or languages.

In Hybrid NER system, the approach is to combine rule-based and machine learning-based methods, and make new methods using strongest points from each method. In this family of approaches Mikheev et. al. proposes a Hybrid document centered system, called LTG system[11], Sirihari et. al. introduce a Hybrid system by combination of HMM, MaxEnt, and handcrafted grammatical rules [6]. Although this type of approach can get better result than some other approaches, but the

weakness of handcraft Rule-base NER remains the same that is when there is a need to change the domain of data.

## 3. Performance Evaluation

### 3.1 Definitions and Scopes

Named Entity is a named object of interest such as a person, organization, or location, its task consists of three subtasks namely, entity names, temporal expressions and number expressions. The expressions to be annotated are unique identifiers of entities (organizations, persons, locations) ENAMEX, times (dates, times) TIMEX, and quantities (monetary values, percentages) NUMEX. The task is to identify all instances of the three types of expressions in each text in the test set and to subcategorize the expressions (ENAMEX, TIMEX, and NUMEX) [1].

### 3.2 Evaluation Metric

Since the system or method must produce a single, unambiguous output for any relevant string in the text, thus, the evaluation is not based on a view of a pipelined system architecture in which Named Entity Recognition would be completely handled as a preprocess to sentence and discourse analysis. The task requires that the system recognize what a string represents, not just its superficial appearance. Sometimes, the right answer is superficially apparent, as in the case of most, if not all, NUMEX expressions, and can be obtained by local pattern-matching techniques. In other cases, the right answer is not superficially apparent, as when a single capitalized word could represent the name of a location, person, or organization, and the answer may have to be obtained using techniques that draw information from a larger context or from reference lists.

A scoring model developed for the MUC and Multilingual Entity Task (MET) evaluations measures both precision (*P*) and recall (*R*), terms borrowed from the information-retrieval community, Where:

$$P = \frac{number\ \ of\ \ correct\ \ responses}{number\ \ of\ \ responses}$$

And

$$R = \frac{number\ \ of\ \ correct\ \ responses}{number\ \ correct\ \ in\ key}$$

These two measures of performance combine to form one measure of performance, the *F*-measure, which is computed by the uniformly weighted harmonic mean of precision and recall:

$$F = \frac{RP}{1/2(R+P)}$$

The term *response* is used to denote "answer delivered by a name-finder", the term *key* or *key file* is used to denote "an annotated file containing correct answers".

In MUC-7, a correct answer from a name-finder is one where the label and both boundaries are correct. There are three types of labels, each of which use an attribute to specify a particular entity. Label types and the entities they denote are defined as follows:

(i) Entity (ENAMEX): person, organization, location.
(ii) Time expression (TIMEX): date, time.
(iii) Numeric expression (NUMEX): money, percent.

A response is half-correct if the label (both type and attribute) is correct but only one boundary is correct. Alternatively, a response is half-correct if only the type of the label (and not the attribute) and both boundaries are correct [1].

### 3.3 Comparison

For comparison, we choose some recent efforts with various methods, where all of them use MUC data set. The MUC data collection was derived from the articles of the air-accidents. The performance of the named entity task is measured by three rates, Recall, Precision, and F (β) that were described in the previous section. We put some results in three tables below. Table 1 shows the results of some method that have used Hand-made method. The results show all systems gave high rate in all parameters. Table 2 indicates results of some systems that have used machine Learning-based methods. The variations in the results were caused by the amount of training datasets and different algorithms. Tables 3 report the results of systems using hybrid methods. In these systems gave high rate in all parameters.

Table 1: Results of experiment with Hand-made Rule NER System

| | System | R | P | $F_{(\beta=1)}$ |
|---|---|---|---|---|
| 1 | IsoQuest,Inc | 90 | 93 | 91.60 |
| 2 | NYU System | 86 | 90 | 88.19 |
| 3 | U. of Manitoba | 85 | 87 | 86.37 |

Table 2: Results of experiment with Machine Learning-based NER System

| | System | R | P | $F_{(\beta=1)}$ |
|---|---|---|---|---|
| 4 | Nymble | N | N | 94.50 |
| 5 | MENE | 89 | 96 | 92.20 |
| 6 | IdentiFinder | 89 | 92 | 90.44 |
| 7 | Support Vector Machine | 89.57 | 83.46 | 86.40 |
| 8 | Association Rule Mining | 66.34 | 83.43 | 70.16 |
| 9 | Maximum Entropy | 43.70 | 60.89 | 50.88 |

Table 3: Results of experiment with Hybrid NER System

| | System | R | P | $F_{(\beta=1)}$ |
|---|---|---|---|---|
| 10 | LTG | 92 | 95 | 93.39 |
| 11 | NYU Hybrid | 85 | 93 | 88.80 |

### 3.4 Results And Discussion

Figure 1 shows however Hand-make approach can get high rate results in specific domain, still it has problem with broad and new domain. Where Hand-make methods are dependent to domain, Machine learning–based methods is the best independent solution for NER. A Comparison between above tables shows that, Machine Learning methods get well result in precision and recall with high portability and it can be best independent and portable solution for text mining and specially NER. But high performance of this kind of methods depends on the data training value. This type of approach can get high precision in recognition when amount of data training is huge, and the result is strictly reduce, when data training value is few or malfunction of algorithm. The Hybrid methods gave good results, but portability of this type of approach is reduced, when they improve precision in recognition by using huge value of fix rules.

## 4. Proposed Method

In this section we introduce our proposed method in NE recognition step, where is a supervised Machine Learning-based method by using Support Vector Machine algorithm. The purpose of Named Entity Recognition approach is converting identification problem into a classification problem and employing a classification statistical model to solve it. In this new approach we will apply fuzzy algorithm to improve classification in Support Vector Machines method, by this way we are going to remove the Support Vector Machines weakness point in multi classification, since in normal classification methods each named entity belongs to a fix class based on its features. We are trying to improve precision in the recognition step in NER method using fuzzy multi classification. We shall use fuzzy algorithm instead of normal classification algorithms, while keeping portability by using machine learning methods. We are going to use of this method in video annotation system to improve searching and indexing in video database systems. The video closed captions, while are in XML forms shall be pass for NER in order to recognize events. The following section briefly describes SVM and our Fuzzy method.

### 4.1 Support Vector Machines

SVM is one of the famous supervised machine learning algorithms for binary classification in all various dataset and it gived the best results where the data set is separable and especially when the training data set is a few, and with extended algorithms it can be used in multi-class problems. To solve a classification task by a supervised machine learning model like SVM, the task usually involves with training and testing data, which consist of some data instances. Each instance in the training set contains one "target value" (class labels, where class label 1 for positive and class label -1 for negative target value) and several "attributes" (features). The goal of a supervised SVM classifier method is to produce a model which predicts target value of data instances in the testing set, when given only the attributes (features). For each SVM, there are two data set namely , training and testing, where the SVM used the training set to makes a classifier model and classify testing data set based on this model with use of their features. Each of the training sample data, is labeled with either positive or negative class tag, as:

$(x_1, y_1)\ldots (Xn, y_n)$, where $x_i \in R^n$, $y_i$ {+1, -1}

That $x_i$ is a feature vector of the $i_{th}$ example represented by and n-dimensional vector. $y_i$ is the label of the $i_{th}$ example, (either +1 for positive or -1 for negative). N is the total number of training examples derived from the training set. (See Figure1).

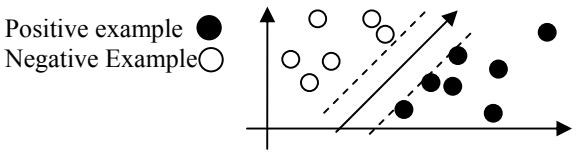


Fig 1. Linear support vector machine classification

In basic form, a SVM learns to find a linear hyperlane that separate both positive and negative examples with maximal margin. This learning bias has proved to have good properties in terms of generalization bounds for the induced classifiers. The maximal margin can be express as follows:

$$(w.\ x) + b = 0, (w \in Rn, b \in R) \qquad (1)$$

The hyperlane separate the training data into positive and negative parts, such that:

$$y_i\ (w.\ x_i) \geqq 1 \quad (2)$$

However, several of such separating hyperlane exists and SVM finds the optimal hyperlane that maximize the margins between the nearest examples to the hyperlane (See Fig 2). The margin ($M$) and the lines can be expressed as:

$$w.\ x + b = \pm 1, M = 2 / ||w|| \quad (3)$$

To maximize this margin is equivalent to minimize the $||\ w\ ||$. This is equivalent to solve the following optimization problem.

Minimize:

$$(1/2)\ \|w\|2 \qquad (4)$$

Subject to: $y_i\ [(w.\ \ x_i) + b] \geqq 1 \qquad (5)$

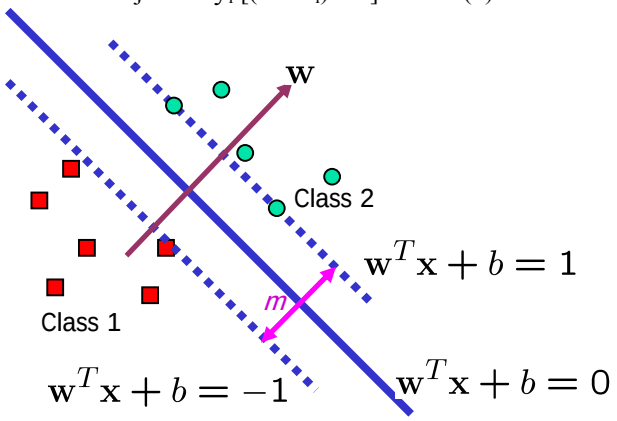


Fig 2. Optimizing hyperplane in linear support vector machine classification

Linear SVM to find a class tag for each data set, it use a sign function as follows:

$$C(x_i) = \text{sign}(w.\ x_i + b)$$

$$c(\mathbf{x}) = \begin{cases} +1 & \text{if } \mathbf{w}\cdot\mathbf{x} + b > 0, \quad \mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}, \\ -1 & \text{otherwise} \end{cases}$$

### 4.2 Fuzzy Named Entity Recognition Method

The first step in our proposed system is to segment the input testing and training data into tokens with a simple tokenizer. In next step, rich feature sets are selected base on the followings.

i) Lexical information (Unigram and Bigram).
ii) Affix (2-4 suffix and prefix letters).
iii) Previous NE information (UniChunk).
iv) Possible NE class.
v) Token feature [2].

In next step we apply our fuzzy member ship function called FSVM to paste a tag to each name (in training and testing) base on below four specifications (See Figure 2), namely

i) Distance to Hyper plane.
ii) Previous named class.
iii) Frequency that the name occurred in this class.
iv) Previous word (Token feature list).

Figure 3 shows our proposed method. In our method each name can get different tag base on this FSVM membership function and instead of fix tag for each name, by this way the system can recognize names semantically.

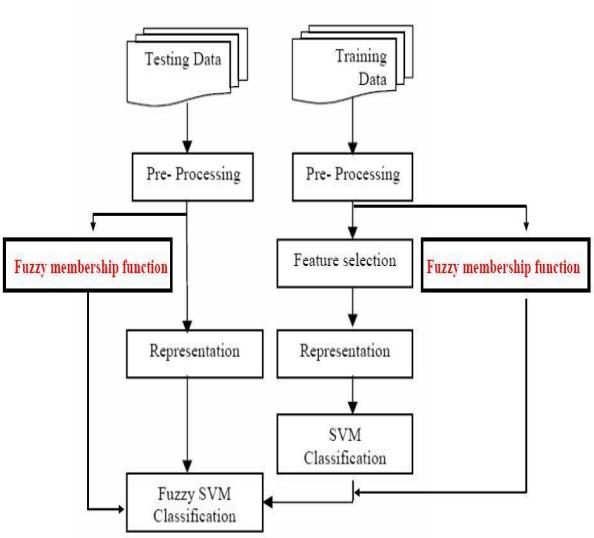


Fig 3. System architecture of the proposed system

In fuzzy membership function in each data set we consider:

$$C(x_i) = \text{sing}(w + b)$$

And FSVM $(x_i)$ as following:

FSVM $(x_i)$ =1 if the $i_{th}$ named belongs to the $j_{th}$ class, For =1, 2,3,4,5

OR

FSVM $(x_i)$ =-1 Otherwise

Fuzzy membership function calculate five marks for each data set that pointed to a class tag and it take a mark for this data set base on four specification that mention above. Range of this value can take a mark between 0 to 100 ranges.

In the next step the system compare this five marks, and the high mark take +1 and this data set is put in this class. By this method class tag is not fixing for names and each name can be recognized dynamically base on meaning and position of name in text or whole document. This method can recognize named entity semantically instead of fix class for each name.

## 5. Conclusion and Future Work

In this paper, we briefly reviewed three types of approach used for Named Entity Recognition. All the proposed methods and models have tried to improve precision in recognition module and portability in recognition domain, as mentioned before, one of the most problems and

difficulties in NER is to change and switch data domain to new domain and that is called portability. In the Rule-based method, there was improvement in precision by adding more rules and developing grammatical rules, however portability was reduce automatically, because of fix rules and methods constructors. We also proposed a new Fuzzy Named Entity Recognition called FSVM to solve second problem in NER, our experimental results with MUC data set show that precision of our method (r=93) is better than traditional SVM method for NER. In future we will improve this fuzzy membership function to recognize names more semantically for QA systems.

## References


[1] Message Understanding Conference, http://www .nlpir.nist.gov/related_projects/muc.

[2] Y.C. Wu, T.K. Fan, Y.S. Lee, S.J Yen, "Extracting Named Entities Using Support Vector Machines", Spring-Verlag, Berlin Heidelberg, 2006.

[3] I. Budi, S. Bressan, "Association Rules Mining for Name Entity Recognition", Proceedings of the Fourth International Conference on Web Information Systems Engineering, 2003.

[4] J. Kim, I. Kang, k. Choi, "Unsupervised Named Entity Classification Models and their Ensembles", Proceedings of the 19th international conference on Computational linguistics, 2002.

[5] F. Bechet, A. Nasr and F. Genet, "Tagging Unknown Proper Names Using Decision Trees", In proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, 2000.

[6] R. Sirhari, C. Niu, W. Li, "A Hybrid Approach for Named Entity and Sub-Type Tagging" Proceedings of the sixth conference on Applied natural language processing ,Acm Pp. 247 - 254 , 2000.

[7] Collins, Michael and Y. Singer. "Unsupervised models for named entity classification", In proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 1999.

[8] D.M. Bikel, S. Miller, R. Schwartz, R, Weischedel, "a High-Performance Learning Name-finder", fifth conference on applied natural language processing, PP 194-201, 1998.

[9] A. Borthwick, J. Sterling, E, Agichtein, and R. Grishman, "Exploiting diverse knowledge sources via maximum entropy in named entity recognition", Proceedings of the Sixth workshop on Very Large Corpora, Montreal, Canada, 1998.

[10] A. Borthwick, J. Sterling, E. Agichtein and R. Grishman, "NYU: Description of the MENE Named Entity System as Used in MUC-7", In Proceedings of the Seventh Message Understanding Conference (MUC-7), 1998.

[11] A. Mikheev, C. Grover, M. Moens, "Description OF THE LTG SYSTEM FOR MUC-7", In Proceedings of the seventh Message Understanding Conference (MUC-7), 1998.

[12] N. Wacholder, R. Yael, C. Misook , "Disambiguation of Proper Names in Text", Proceedings of the 5th Applied Natural Language Processing Conference, 1997.

[13] D. Appelt, and et. al., "SRI International FASTUS system MUC-6 test results and analysis", Proceedings of the MUC-6, NIST, Morgan-Kaufmann Publisher, Columbia, 1995.

[14] L. Iwanska, M. Croll, T. Yoon, and M. Adams, "Wayne state university: Description of the UNO processing system as used for MUC-6", In Proc. of the MUC-6, NIST, Morgan-Kaufmann Publishers, Columbia, 1995.

[15] Morgan, R., and et. al., "University of durham: Description of the LOLITA system as used for MUC-6" In Proc of the MUC-6, NIST, Morgan-Kaufmann Publishers, Columbia, 1995.

[16] R. Grishman, "The NYU System for MUC-6 or Where's the Syntax", In Proceedings of the Sixth Message Understanding Conference (MUC-6), 1995.

[17] D. Appelt, and et. al., "FASTUS: A finite state processor for information extraction from real-world text", Proceedings of IJCAI, 1993.


**Alireza Mansouri** is a MSc. student in Computer Science at The University Putra Malaysia. He received the B.E. degree from Lahijan Azad Uniyersity in 1996. His research interests include Information Extraction and Information Retrieval, Data Mining, Video database and Machine Learning.

**Dr. Lilly Suriani Affendey** is a lecturer in Department of Computer Science, Faculty of Computer Science and Information Technology, University Putra Malaysia, Serdang. She received her B.E. from UPM in 1991. She received her MSc. from university of Bradford, United Kingdom in 1994. She obtained her Ph.D. from UPM in 2007. Her research interests include multimedia database, data mining, and intelligent computing.

**Dr. Ali Mamat** is an associate professor in the Department of Computer Science, Faculty of Computer Science and Information Technology, University Putra Malaysia ,Serdang. He Obtained his Ph.D. in Computer Science from University of Bradford, U.K. in 1992. He has published more than 50 papers in international journals and proceedings. His research interests include databases, XML storage and web semantics.