Hierarchical, Perceptron-like Learning for Ontology-Based Information Extraction

Yaoyong Li
University of Sheffield
211 Portobello Street
Sheffield, S1 4DP, UK
yaoyong@dcs.shef.ac.uk

Kalina Bontcheva
University of Sheffield
211 Portobello Street
Sheffield, S1 4DP, UK
kalina@dcs.shef.ac.uk

ABSTRACT

Recent work on ontology-based Information Extraction (IE) has tried to make an increased use of the knowledge from the target ontology in order to improve the semantic annotation results. However, only very few approaches are able to benefit from the ontology structure and one of them is not a learning system, thus is not easy to adapt to new domains, whereas the other one does not perform semantic annotation of documents, but only ontology population.

This paper introduces a hierarchical learning approach for IE, which uses the target ontology as an essential part of the extraction process. Hierarchical classification takes into account the relations between concepts, thus benefiting directly from the ontology.

We also carry out evaluation experiments on the largest available semantically annotated corpus of 146 classes. The results demonstrate clearly the benefits of using knowledge from the ontology for ontology-based IE. We also demonstrate the advantages of our approach over other state-of-the-art learning systems on a commonly used benchmark dataset.

Keywords

Ontology-based Information Extraction, semantic annotation, hierarchical learning

1. INTRODUCTION

At present there is a large gap between Knowledge Management (KM) systems and the natural language materials that form something approaching 80% of corporate data stores [22]. Similarly, Gartner reported in 2002[1] that for at least the next decade more than 95% of human-to-computer information input will involve textual language. They also report that by 2012 taxonomic and hierarchical knowledge mapping and indexing will be prevalent in almost all information-rich applications. There is a tension here: between the increasingly rich semantic models in KM systems on the one hand, and the continuing prevalence of human language materials on the other. The process of tying semantic models and natural language together is referred to as *semantic annotation*. This process may be characterised as the dynamic creation of interrelationships between *ontologies* and unstructured and semi-structured documents in a bidirectional manner.

Information Extraction (IE), a form of natural language analysis, is becoming a central technology for bridging the gap between unstructured text and formal knowledge expressed in ontologies. Ontology-Based IE (OBIE) is IE which is adapted specifically for the semantic annotation task. One of the important differences between traditional IE and OBIE is in the use of a formal ontology as one of the systems resources and may also involve reasoning. Some researchers [20] call ontology-based any system which specifies its outputs with respect to an ontology, however, in our view, if a system only has a mapping between the IE outputs and the ontology, this is not sufficient and we refer to such systems as *ontology-oriented*.

Another substantial difference of the semantic IE process from the traditional one is the fact that it not only finds the (most specific) class of the extracted entity, but it also identifies it, by linking it to its semantic description in the instance base. This allows entities to be traced across documents and their descriptions to be enriched during the IE process. In practical terms, this entails named entity and relation extraction and also coreference resolution both within and across documents.

Recent work on ontology-based IE (see Section 2) has tried to make an increased use of the knowledge from the target ontology in order to improve the semantic annotation results. However, only very few approaches are able to benefit from the ontology structure and one of them is not a learning system, thus is not easy to adapt to new domains, whereas the other one does not perform semantic annotation of documents, but only ontology population.

The main contribution of this paper is to investigate the application of hierarchical classification learning for semantic annotation, as part of an ontology-based IE system. Hierarchical classification takes into account the relations between concepts, thus benefiting directly from the ontology. In particular, this paper studies the large margin hierarchical classification learning algorithm Hieron proposed in [9], because it is very efficient during both training and classification. Computational efficiency is of major importance for OBIE, because depending on the size of the ontology, the system may need to train hundreds of classifiers.

However, it should be noted that work presented in this paper is not a simple application of the Hieron algorithm proposed in [9] to OBIE. In fact, the OBIE requirements lead to several important modifications – introduction of multi-loop learning and a parameter which ensures that the

[1] http://www3.gartner.com/DisplayDocument?id=379859

algorithm is applicable to any IE corpus. Both of these resulted in a quantifiable improvement in performance (see Table 6 in Section 5). In addition, the semantic annotation task is very different from conventional document classification, which is what Hieron was originally developed for. Consequently, another contribution of this work is in showing how OBIE can be decomposed into several hierarchical classification tasks, which can then be approached with an algorithm such as Hieron.

Another important contribution of this work is in the use of the Sekt[2] ontology-annotated news corpus for detailed evaluation of our system. To the best of our knowledge, it is the only corpus suitable for evaluating OBIE, which has more than 20 classes (146 classes) from a non-trivial ontology (Proton).

The problem with using only the Sekt corpus is that no other systems have been evaluated on it, apart from the SVM and Perceptron ones reported here. Unfortunately other recent corpora for IE evaluation (e.g., Pascal challenge[3], CONLL'03[4]) are either not fully available or use a small, flat set of labels (fewer than 20), thus making them inappropriate for experimenting with semantic annotation on a realistic scale.

The experimental results demonstrate that our hierarchical classification algorithm obtains clearly better results than SVM and Perceptron both in terms of the conventional precision and recall IE metric and an ontology-induced metric. In addition, in order to provide some comparison against state-of-the-art learning IE systems, we also evaluate our Hieron-based system on the seminar corpus, where it achieves the best overall performance.

The paper is structured as follows. Related work on ontology-based IE is discussed in Section 2 and our work is placed in that context. Section 3 introduces the large margin hierarchical classification algorithm Hieron, our modifications to it, the hierarchy-based cost measure, which Hieron requires, and the way in which the OBIE task is formalised as a classification problem. The Sekt ontology-annotated gold-standard is described next (section 4), followed by a comprehensive set of experimental results and their analysis (section 5). We also compare our Hieron-based learning approach against other state-of-the-art IE learning algorithms. The paper concludes with a discussion and future work plan.

2. RELATED OBIE SYSTEMS

There are a number of what we call ontology-oriented IE systems, which, unlike ontology-based ones, do not incorporate ontologies into the system, but either use them as a bridge between the IE output and the final annotation (as with AeroDAML) or rely on the user to provide the relevant information through manual annotation (as with the Amilcare-based tools).

AeroDAML [17] applies IE techniques to automatically generate DAML annotations from web pages. it links most proper nouns and common types of relations with classes and properties in a DAML ontology. AeroDAML uses an ontology which is used to translate the extraction results into a corresponding RDF model.

Amilcare [7] is an IE system which has been integrated in several different semantic annotation tools. It uses supervised rule learning to adapt to new domains and applications given human annotated texts (training data). It treats the semantic annotations as a flat set of labels, thus ignoring the further knowledge in the ontology. Similar to us, Amilcare uses GATE's NLP components [8] in order to obtain linguistic information as features for the learning process. Amilcare has formed the basis of several semantic annotation tools – MnM [21] and S-CREAM [15] – which combine a manual annotation tool and the IE system in the background.

One of the problems with these annotation tools is that they do not provide the user with a way to customise the integrated language technology directly. While many users would not need or want such customisation facilities, users who already have ontologies with rich instance data will benefit if they can make this data available to the IE components. However, this is not possible when "traditional" IE methods like Amilcare are used, because they are not aware of the existence of the users ontology.

The more serious problem however, as discussed in the S-CREAM system [15], is that there is often a gap between the IE output annotations and the classes and properties in the users ontology. The proposed solution is to write some kind of rules, such as logical rules, to resolve this. For example, an IE system would typically annotate London and UK as locations, but extra rules are needed to specify that there is a containment relationship between the two (for other examples see [15]). However, rule writing of the proposed kind is too difficult for most users and therefore ontology-based IE is needed, as it annotates directly with the classes and instances from the user's ontology.

In response to these problems, a number of OBIE systems have been developed. Magpie [11] is a suite of tools which supports semantic annotation of web pages. It is fully automatic and works by matching the text against instances in the ontology. The SemTag system [10] is similar in approach to MagPie as it annotates texts by performing lookup against the TAP ontology. It also has a second, disambiguation phase, where SemTag uses a vector-space model to assign the correct ontological class or determine that this mention does not correspond to a class in TAP. The problem with both systems is that they are not able to discover new instances and are thus restricted in terms of recall.

The PANKOW system [5] exploits surface patterns and the redundancy on the Web to categorise automatically named entities found in text with respect to a given ontology. The main disadvantage of this approach is that it does not compare the context in which the proper name occurs in the document to the contexts in which it occurs on the Web, thus requiring human input in order to disambiguate instances with the same name that belong to different classes in different contexts (e.g., Niger can be a river or country).

OntoSyphon [20] is similar to PANKOW and uses the ontology as the starting point and carries out web mining to populate the ontology with instances. It uses the ontology structure to determine the relevance of the candidate instances. However, it does not carry out semantic annotation of documents, which is the problem addressed here.

The KIM OBIE system [16] produces annotations linked both to the ontological class and to the exact individual in the instance base. For new (previously unknown) entities,

[2]For further information on the sekt project, see http://www.sekt-project.com. The corpus itself is available on request from the second author.

[3]http://nlp.shef.ac.uk/pascal/Corpus.html

[4]http://www.cnts.ua.ac.be/conll2003/ner/

new identifiers are allocated and assigned; then minimal descriptions are added to the semantic repository. KIM has a rule-based, human-engineered IE system, which uses the ontology structure during pattern matching and instance disambiguation. The only shortcoming of this approach is that it requires human intervention in order to adapt it to new ontologies.

To summarise, all these systems use the ontology as their target output and the ontology-based ones also use class and instance information during the IE process. While KIM and OntoSyphon do make use of the ontology structure, the former is a rule-based, not a learning approach, whereas the latter does not perform semantic annotation, only ontology population.

The next section discusses how the ontology structure can be exploited, in addition to class and instance information, within a machine learning approach to ontology-based information extraction.

3. EXPLOITING THE ONTOLOGY

The categories of information entities in conventional IE have no specific relation among them, i.e., they are treated as independent of each other by the learning algorithms. In contrast, as concepts in an ontology are related to each other (at the very least through the subsumption hierarchy), it is beneficial to exploit this knowledge in OBIE.

This section exploits two aspects of using the ontology structure for OBIE. First it discusses ontology-induced measures, which are then used by the learning algorithm, in addition to calculating some distance-based metrics. Secondly, it introduces the Perceptron-based learning algorithm Hieron which has a mechanism to handle effectively hierarchical classification and our adaptations of Hieron for the OBIE requirements.

3.1 Ontology-Based Performance Metric

As concepts in ontologies are related to each other in a subsumption hierarchy, the cost (or loss) for an instance of a concept X being wrongly classified as belonging to another concept Y is defined as dependent on the two particular concepts (denoted as $c(X, Y)$). Given this cost measure, one OBIE requirement is that if the system misclassifies a mention in the text as belonging to class Y, instead of X, the cost of that misclassification should be as small as possible, (e.g., classify it as a super-class of the correct class).

IE systems traditionally use evaluation metrics, such as *precision*, *recall* and F_1, which are computed for each category independent of all other categories. An overall performance measure is obtained by averaging the performances for all categories (namely macro-averaged) or by putting together all the results of all classifications (micro-averaged). However, this type of measures do take into account the hierarchical relations between classes in the ontology and their associated misclassification costs. Consequently, another OBIE requirement is to have performance metrics which are sensitive to the structure of the target ontology. Therefore, next we generalise the commonly used IE metrics, *precision*, *recall* and F_1 to ontology-based ones.

In order to evaluate an OBIE system on a corpus annotated with a given ontology, we first compute the following three numbers on the system-annotated corpus:

- n — number of mentions which are instances of concepts in the ontology and have been found by the OBIE system (regardless of whether or not OBIE assigned them the correct class).
- $n_{missing}$ — number of entities in the corpus which are instances of concepts in the ontology but are not found by the system.
- $n_{spurious}$ — number of the entities recognised by the system which actually are not an instances of any concept in the ontology.

Then each pair of concepts X and Y we define the cost measure $c(X, Y)$ as a non-negative number, equal to 0 if $X = Y$. If we assume that C is the largest cost for a given ontology, then we can define a cost based error as $e_{cost}(X, Y) = c(X, Y)/C$, satisfying that $e_{cost}(X, Y) \in [0, 1]$ and $e_{cost}(X, Y) = 0$ if $X = Y$.

Based on the cost-based error, an overall accuracy for the n entities identified by the system is defined as follows:

$$a_{cost} = \sum_{i=1}^{n}(1 - e_{cost}(A_i, B_i)) \quad (1)$$

where A_i and B_i are two classes in the ontology and $e_{cost}(A_i, B_i)$ is the cost of misclassifying the ith entity as an instance of class B_i, instead of its correct class A_i (class B_i is the same as A_i if the ith entity is classified correctly).

Using the overall accuracy a_{cost} we define ontology-based precision and recall, respectively, as

$$P_o = a_{cost}/(a_{cost}+n_{spurious}), \quad R_o = a_{cost}/(a_{cost}+n_{missing})$$

Then, as with conventional f-measure, the ontology-based F_1 is defined as the harmonic mean of ontology-based precision and recall:

$$F_{o1} = 2 * P_o * R_o/(P_o + R_o) \quad (2)$$

In other words, ontology-based F_{o1} is a generalisation of the standard F_1. In fact, if we define the cost $c(X, Y)$ as the binary function

$$c(X, Y) = \begin{cases} 0 & \text{if } X = Y \\ 1 & \text{otherwise} \end{cases} \quad (3)$$

then F_{o1} would be reduced to the standard overall F-measure.

Recent studies of hierarchical classification (see e.g. [9, 2]) typically define $c(X, Y)$ as the distance $\gamma(X, Y)$ between the two nodes X and Y in the classification hierarchy. In our experiments we used the distance between two classes in the ontology as their misclassification cost and henceforth this is referred to as the distance-based metric.

3.2 Hierarchical Learning with Hieron

The *Hieron* large margin learning algorithm for hierarchical classification was defined by [9], based on the margin Perceptron algorithm. Hierarchical classification refers to a specific multi-class classification problem where the class labels are organised in a hierarchical fashion. One example is document categorisation where categories belong to a taxonomy. Here we briefly describe the learning algorithm and discuss our modifications to the original algorithm and then discuss how to apply it to OBIE.

As defined, the Hieron algorithm exploits the hierarchical structure of the class labels. It learns one Perceptron model for each class and meanwhile ensures that the difference between two models is in proportion to the distance of the two

classes in the tree. The philosophy of the learning algorithm is that, if we have to misclassify one example as class X instead of Y, then that class X should be close to the correct class Y in the hierarchical structure.

Let us suppose a hierarchical classification problem which has instance domain $\mathcal{X} \subseteq \mathbb{R}^n$ and label set $\mathcal{Y}$. The labels in the set $\mathcal{Y}$ can be arranged as nodes in a rooted tree $\mathcal{T}$. For any pair of labels $u, v \in \mathcal{Y}$, let $\gamma(u, v)$ denote their distance in the tree, namely the number of edges along the (unique) path from u to v in $\mathcal{T}$. For every label v in the tree, $\mathcal{P}(v)$ is defined as the set of labels along the path from the root to v, inclusive.

The training set $\mathcal{S} = \{(\mathbf{x}_i, y_i) : i = 1, \ldots, m\}$ contains instance-label pairs, where each $\mathbf{x}_i \in \mathcal{X}$ and each $y_i \in \mathcal{Y}$. The Hieron learning algorithm aims to learn a classification function $f : \mathcal{X} \to \mathcal{Y}$ which has a small tree induced error. The classifier f has the following form: each label $v \in \mathcal{Y}$ has a matching prototype $\mathbf{W}^v \in \mathbb{R}^n$, and the classifier f makes its predictions according to the following rule:

$$f(\mathbf{x}) = \text{argmax}_{v \in \mathcal{Y}} \langle \mathbf{W}^v, \mathbf{x} \rangle \tag{4}$$

where $\langle \cdot, \cdot \rangle$ represents the inner product of two vectors. Hence, the task of learning f is reduced to learning a set of prototypes $\{\mathbf{W}^v : v \in \mathcal{Y}\}$.

However, Hieron does not deal directly with the set of prototypes but rather with the difference between the prototype of each node and the prototype of its parent. Formally, we denote $\mathcal{A}(v)$ as the parent node of v in the tree and assume that the parent node of a root node is the root itself. Then the difference weight vector is defined as $\mathbf{w}^v = \mathbf{W}^v - \mathbf{W}^{\mathcal{A}(v)}$. Each prototype is now decomposed into the sum

$$\mathbf{W}^v = \sum_{u \in \mathcal{P}(v)} \mathbf{w}^u \tag{5}$$

Since the learning algorithm requires that adjacent vertices in the label tree have similar prototypes, by representing each prototype as a sum of vectors from $\{\mathbf{w}^v : v \in \mathcal{Y}\}$, adjacent prototypes $\mathbf{W}^v$ and $\mathbf{W}^{\mathcal{A}(v)}$ can be kept close by simply keeping the norm of the weight vector $\mathbf{w}^v = \mathbf{W}^v - \mathbf{W}^{\mathcal{A}(v)}$ small.

The Hieron algorithm assumes that there exists a set of weight vectors $\{\boldsymbol{\omega}^v : v \in \mathcal{Y}\}$ such that the following hold:

$$\sum_{v \in \mathcal{P}(y_i)} \langle \boldsymbol{\omega}^v, \mathbf{x}_i \rangle - \sum_{u \in \mathcal{P}(r)} \langle \boldsymbol{\omega}^u, \mathbf{x}_i \rangle \geq \sqrt{\gamma(y_i, r)}, \tag{6}$$
$$\forall (\mathbf{x}_i, y_i) \in \mathcal{S} \text{ and } \forall r \in \mathcal{Y} \backslash \{y_i\}$$

However, note that this assumption can be relaxed if we introduce some regulation parameter into the learning algorithm, as discussed below.

The Hieron learning algorithm is similar to the Perceptron algorithm as it learns one classifier per class. But, unlike Perceptron where each model is learned independently of the others, it learns the Perceptrons for all classes in a collective way. The algorithm initialises each of the Perceptron's weight vectors $\{\mathbf{w}^v : v \in \mathcal{Y}\}$ as a zero vector and updates a weight vector only if a prototype related with it makes a wrong prediction. By doing so the learning algorithm tries to keep the norm of the weight vector small, which is one of the requirements as discussed above.

The learning algorithm also tries to satisfy the margins requirement for the weight vectors and training set shown in (6). Formally, for each instance-label pair $(\mathbf{x}_i, y_i) \in \mathcal{S}$, the learning algorithm checks if the current weight vectors satisfy the margin requirement for each label $y \neq y_i$ by computing the following loss function:

$$L(\{\mathbf{w}^v\}, \mathbf{x}_i, y_i, y) = \sum_{u \in \mathcal{P}(y)} \langle \mathbf{w}^u, \mathbf{x}_i \rangle - \sum_{v \in \mathcal{P}(y_i)} \langle \mathbf{w}^v, \mathbf{x}_i \rangle + \sqrt{\gamma(y_i, y)} \tag{7}$$

The margin requirement for $(\mathbf{x}_i, y_i)$ and y is satisfied if and only if the above function is less than or equal to 0. If the margin requirement is satisfied for all training examples, then the learning stops and returns the current learned model. Otherwise, from all training examples $(\mathbf{x}_i, y_i)$ for which the margin requirement (6) is violated by the current model, choose the label $\hat{y}_i$ that violate the margin requirement the most (namely it has the maximal value of the function (7)), and update the current weight vectors comprising the two prototypes $\mathbf{W}^{y_i}$ and $\mathbf{W}^{\hat{y}_i}$, respectively, as illustrated in Figure 1.

As shown in Figure 1, when a training example $\mathbf{x}$ with label y is predicted mistakenly as label y', only the weight vectors associated with the nodes in the shortest path linking nodes y and y' are updated, except for the MSCA node. In other words, only the nodes depicted using solid lines are updated, in which the symbol '+' means increasing the corresponding weight vector by the example $\mathbf{x}$ and the symbol '-' means decreasing the weight vector by $\mathbf{x}$.

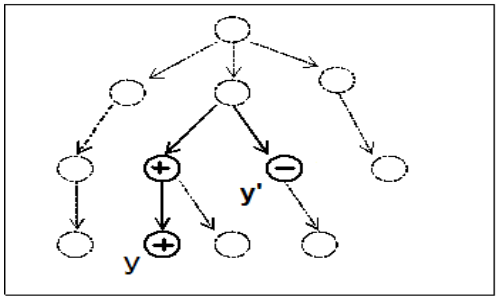

Figure 1: Illustration of the Hieron's update.

As already discussed above, in order to ensure that adjacent vertices in the label tree have similar prototypes, the Hieron algorithm needs to keep the norms of the weight vector $\mathbf{w}$ as small as possible. This is achieved by initialising all weight vectors to zero and only updating them if necessary.

3.3 Our Hieron Modifications

The learning algorithm described above is the original Hieron batch learning algorithm presented in [9]. In order to achieve better performance, some modifications were introduced in our implementation, as follows:

1. Our learning algorithm learns from the training set until no error is made on the training examples, which means that more than one learning loop may be needed on the training set. In contrast, the original Hieron batch learning just allowed one learning loop. It will be shown by our experiments described below that multi-loop learning has better generalisation performance than single loop learning.

2. The Hieron learning algorithm requires that the training set is compatible with the margin conditions de-

scribed in equation (6), so that the algorithms stops after a finite number of loops. This is a problem because in OBIE often it is not known in advance whether or not a training set satisfies the margin condition. Therefore, we introduce a regulation parameter into the algorithm such that the learning would stop after some loops on any training set. The value of the regulation parameter is a positive number[5]. The regulation parameter is similar to that used for Perceptron (see e.g. [19]). In more detail, the role of the regulation parameter is to add an extra dimension to the feature vector of each training example and set the component of the dimension as the regulation parameter. By doing so the set of the training examples with extended feature vectors would become linearly separable regardless of the linear separability of the original training set, meaning that the Hieron algorithm would stop after a finite number of loops for any training data.

In addition, the original algorithm [9] distinguishes two types of learning models. One type was the weight vectors obtained at the end of learning, namely $\{\mathbf{w}_i^m : v \in \mathcal{Y}\}$. Another was the mean of all weight vectors used during learning. Let us assume that we apply the weight vectors m times to training examples during learning and the weight vectors used were $\{\mathbf{w}_i^v : v \in \mathcal{Y}, i = 1, \cdots, m\}$, then for every $v \in \mathcal{Y}$ define the means of weight vectors as

$$\mathbf{w}^v = \frac{1}{m} \sum_{i=1}^{m} \mathbf{w}_i^v \tag{8}$$

It was shown in [9] that the averaged weight vectors had better results than the last weight vectors in most cases. In our OBIE experiments we also compare the two types of weight vectors (see Section 5).

3.4 The Hieron-based OBIE System

As already discussed, the goal of OBIE is to identity and classify information entities in text as instances of concepts in an ontology. On the other hand, the Hieron is basically a classification algorithm which classifies every example into categories organised in a tree structure. In order to apply the Hieron to OBIE, we need to adapt the OBIE task to make it similar to the hierarchical classification problem.

First, we convert the OBIE task into two hierarchical classification problems. Traditionally, when classifiers are used for information extraction each token in the text is treated as one example and classified as belonging to one of the target IE labels or as having no label (see e.g., [13, 18]). In particular, the annotation task can be viewed as two binary classification problems, one is for recognising the start tokens of information entities and the other one is for the end tokens. Similarly, we transform the OBIE task into two hierarchical classification problems. For each class in the ontology, two hierarchical classifiers are trained – one for recognising the start token of the class instances and one for the end.

Secondly, as there are tokens in the text which do not belong to any class in the ontology, in order to apply Hieron to OBIE, the ontology is extended notionally with a new child node of the root node, that represents the concept of non-relevant token. However, this added concept is not considered in the evaluation metrics, as it is only required for the proper functioning of the classification algorithm.

Thirdly, note that the Hieron algorithm requires that classes are organised in a tree. However, for some ontologies, the class subsumption hierarchy is not a tree, but a graph where some concepts occur as subclasses of two or more classes. In other words, some nodes in the ontology may have more than one parent. The Proton ontology used in our experiments (see below) is one example of this kind of ontology. For example, the concept *SportBuilding* is a sub-concept of *Building* and *SportFacility*.

Consequently, the Hieron algorithm had to be adapted to deal with graph-like hierarchies, such as the Proton ontology. The modification we made on Hieron was simple. We compute one prototype vector for each path from the root node to the given class using formula (5). Then, given one example during training or application, the inner products between that example and every prototype vectors are compared with each other and the example is assigned the class whose prototype is most relevant.

Finally, we replace the distance $\gamma(X, Y)$ in the Hieron algorithm with the cost measure $c(X, Y)$ between two concepts in the ontology. Therefore, we can learn classifiers which are optimised according to a distance-based cost measure, which encodes structural knowledge from the ontology.

4. EXPERIMENTAL DATASET

The corpus used in our experiments consists of semantically annotated news articles. The articles were divided into three subsets according to the article's theme, namely business, international politics and UK politics, which has 91, 99 and 100 articles, respectively. The corpus was annotated manually according to the Proton ontology[6]. Our experiments described below used the *psys:Entity* branch of the Proton ontology that has 272 unique concepts.

As already discussed above, some concepts in Proton have more than one parent class and the hierarchical structure of Proton has 10 levels, with maximal path length 14.

The news corpus was annotated with 167 concepts, of which 146 concepts are in the Proton ontology and are used in our experiments[7]. The concepts span from the 3rd to the 10th level of the hierarchical structure of Proton. As the corpus was created in the Sekt project, hereafter we will refer to the corpus as the Sekt ontology-annotated news corpus.

Table 1: Distribution of concepts with different numbers of instances in the Sekt ontology-annotated corpus.

#instances	1	2	3	4	5	6 – 10	11 – 20	>20
#concepts	23	14	12	6	2	13	21	55

Table 1 presents the distribution of concepts with different numbers of instances in the corpus. We can see that there are 57 concepts each of which has 5 instances or less, thus presenting a data sparseness problem.

[5]Although the optimal value of regulation parameter may be dependent on the data used, in our experiments described in this paper, we simply set the regulation parameter as 1.

[6]See http://proton.semanticweb.org.

[7]The other 21 concepts were proposals as extensions to the ontology, but have not been adopted yet and therefore are not used in these experiments.

Table 2: Numbers of instances of the seven concepts in the three parts of the Sekt ontology-annotated corpus.

	#Doc	Person	Loc	Org	Money	Contact	Temporal	Time
Business	91	336	601	1385	490	2	119	743
International	99	948	1857	840	75	1	112	550
UK	100	845	847	845	100	0	106	670

Table 3: Example features for the text "Time: 3:30 PM". The Unknown value for the word "Time" means that ANNIE identified the word "Time" as a named entity but could not attribute a more specific entity type.

Token	Case	Tokenkind	Lemma	Pos	Lookup	ANNIE Entity
Time	upperInitial	word	time	NNP		Unknown
:		punctuation	:	:		
3		number	3	CD		Time
:		punctuation	:	:		Time
30		number	3	CD		Time
PM	allCaps	word	pm	NNP	time	Time

In order to examine the effect of data sparseness on the algorithm's performance, we also created a version of the corpus where all classes were generalised into 7 high-level classes, which are broadly equivalent to labels used in traditional IE systems (e.g., Person, Location, etc). Table 2 presents the numbers of instance of each of the seven concepts in each part of the corpus.

The corpus is pre-processed with the open-source ANNIE system, which is part of GATE [8]. This enabled us to use a number of linguistic features, in addition to information already present in the document such as words and capitalisation information. The linguistic features are domain-independent and include token kind (word, number, punctuation), lemma, part-of-speech (POS) tag, gazetteer class, and named entity type according to ANNIE's rule-based name entity recogniser[8]. Table 3 shows an example of text with its associated features. Note that a token may not have values for all features, e.g. the token "Time" does not have value for the Lookup feature because it does not occur in ANNIE's gazetteer lists.

Since in IE the context of the token is usually as important as the token itself, the learning algorithm takes into account features of the preceding and following tokens, in addition to those of the token being classified. In our experiments the same number of left and right tokens was taken as a context window. In the experiments discussed below, the window size is 4, which means that the algorithm uses features derived from 9 tokens: the 4 preceding, the current, and the 4 following tokens. Due to the use of a context window, the input vector is the combination of the feature vector of the current token and these of its neighboring tokens.

5. EXPERIMENTAL RESULTS

The experiments reported here evaluate our modified implementation of the Hieron learning algorithm on the Sekt ontology-annotated news corpus. As there are no previously reported results on this corpus, we compare Hieron against two state-of-the-art "traditional" learning algorithms for IE: SVM and Perceptron. Since Hieron exploits the relationships among classes in the ontology, the expectation is that it would perform better on OBIE than SVM and Perceptron which were designed basically for flat classification. In other words, these algorithms ignore the relationships between the classes and treat them as independent of each other.

As already discussed in section 3.2, the Hieron algorithm is very similar to the uneven margins Perceptron except that Hieron takes into account the relationship among classes as well as the misclassification cost measure $c(X, Y)$.

In these experiments, we used the uneven margins SVM and Perceptron with uneven margins (PAUM), instead of the standard algorithms, because the uneven margins algorithms have better performance than the respective standard models for IE (see [18]).

Table 4: Comparison of the performance of Hieron, SVM and Perceptron learning on OBIE: micro-averaged F_1 (%) and distance induced F_{o1}(%).

	Micro-averaged F_1			Distance induced F_{o1}		
	PAUM	SVM	Hieron	PAUM	SVM	Hieron
Bu.	74.1	75.3	**82.7**	78.8	79.3	**91.2**
Int	77.1	80.1	**83.3**	83.0	85.9	**91.3**
UK	82.0	**82.9**	82.5	83.6	84.4	**90.1**

Table 4 presents the results of the three learning algorithms on the Sekt ontology-annotated news corpus, measured both by conventional micro-averaged F_1 and the distance-based F_{o1} defined in formula (2). Recall that the micro-averaged f-measure treats concepts in ontology as independent, whereas the distance-based metric takes into account the relationships in the ontology. In other words, if a mention in the text is found, but assigned mistakenly the wrong ontology class, then no credit is given with the conventional micro-averaged F1, but some credit (which is inversely proportional to the distance between the two concepts in ontology) will be given by the distance-based measure.

For each algorithm, we run three experiments which use one of the three subsets of the corpus for testing and the other two for training.

Firstly, as shown in Table 4, Hieron outperforms the non-ontology-based IE methods measured by conventional F_1.

[8]These entity labels are: Person, Organisation, Location, Date, Money, and Address.

The 5 to 10% improvement in results demonstrates that the IE algorithm does benefit from taking into account the relationships between classes in the ontology and using this information during the learning process to optimise performance.

In addition, Hieron consistently achieves significantly higher performance than SVM and Perceptron according to the distance-based F_{o1}, which gives partial credit for ontological "near misses". This increased performance is due to the optimisation mechanism built into Hieron, where the distance between the classifiers for two concepts is proportional to the distance of the two concepts in the ontology. In contrast, SVM and other traditional IE learning methods are trained to treat each concept independently from the rest. Consequently, when Hieron misclassifies a mention, it is much more likely than the flat algorithms to assign a close concept if it cannot identify the correct one exactly Consequently, this is the reason for the big difference in performance between the two metrics.

Table 5 compares the computation times of training and application of the three learning algorithms on the ontology corpus. The business and international politics parts of the corpus were used for training and the UK politics part for testing. We ran those experiments on a Linux server with one Pentium 4 CPU (3.0GHz) and 2G RAM. The results show that Perceptron is very fast, whereas SVM takes much longer than both Perceptron and Hieron, particularly for application. At the same time, its performance in all experiment is only slightly better than Perceptron.

The Hieron algorithm takes longer than Perceptron, but performs significantly better than both traditional IE algorithms. As the test set consisted of 100 documents, the Hieron effectively took on average 1 second per document.

Table 5: Training and test times (in second) of the learning algorithms: Perceptron, SVM, and Hieron.

	PAUM	SVM	Hieron
Training	552s	20528s	3815s
Test	33s	11187s	109s

Table 6: Comparisons of the three variants of the Hieron: micro-averaged F_1 (%) and distance induced F_{o1} (%). Training time is shown in second.

	Single loop		Multi-loop		Regulation	
	Last	Mean	Last	Mean	Last	Mean
MA	79.8	79.1	81.3	82.2	**82.5**	**82.5**
Dist	89.0	88.3	89.3	89.8	**90.1**	90.0
time	510s	989s	54173s	97460s	4124s	11416s

As already discussed in Section 3.2, we modified the Hieron algorithm presented in [9] by introducing multiple learning cycles on the training set. We also introduced a regulation parameter to each weight vector to guarantee that the training would finish after a finite number of learning cycles on any training corpus. Table 6 compares the performance of the original Hieron version against our modified algorithm, using the business and international politics subsets for training and the UK politics subset for testing. Results for the last weight as well as the mean of all obtained weights during learning are reported for all three Hieron variants.

Table 7: Comparison of the exact match and partial match for the SVM and Hieron: conventional micro-averaged F_1 (%) and the distance based F_{o1} (%).

	F_1		Distance F_{o1}	
	SVM	Hieron	SVM	Hieron
Exact	79.3	76.7	80.5	82.0
Exact and partial	82.9	82.5	84.4	90.1

As shown in Table 6, multi-loop learning (300 loops used in the experiment) on its own achieves better results than single loop learning, in particular with respect to the conventional micro-averaged F_1, showing that multi-loop learning can exploit better the data regularity than the single loop can.

Secondly, when the regulation parameter is added to the multi-loop learning the performance improves even further, while the training time of the former is only about one fourteenth of the time of the latter.

Finally, averaged weights perform slightly better than the last weight in some cases and a bit worse in the other cases. However, this also requires much higher computation time than the last weights, so there is a trade-off between performance and training time which means that for practical systems using last weight would probably be more useful.

Last but not least, it should be noted that the differences in computation times between these variants of the Hieron algorithm affect only training and application times are the same.

5.1 Scoring Partially Correct Results

When IE systems are evaluated the entities predicted by the system are compared to the entities in the human-annotated gold standard. In the above experiments we used Hieron to recognise separately the start and end tokens of the instances. Therefore, if both the start and end boundaries are recognised correctly, we say the entity is recognised correctly or is an exact match to the gold standard. On the other hand, if only the start or end token match those from the gold standard, then it is referred to as a partial match or a partially correct result.

Few IE systems (e.g. [6]) are evaluated by using exact match only and partially correct results are discarded. However, traditional large-scale evaluations of IE systems, e.g., the Message Understanding Conferences (MUC) give also credit to partial matches [4]. Consequently, in the experiments reported above, we took into account both correct and partially correct results and, again following traditional IE scoring methods, partial results are assigned half the score of an exact match.

Table 7 compares the results for exact match against those for correct and partial match combined. For exact match Hieron performs worse than SVM with respect to conventional f-measure. However, if partially correct results are taken into account, Hieron's results are higher, showing that the hierarchical classifier has better capability than the "flat" SVM IE in recognising the presence of instances in the text, even when it is not able to recognise their boundaries exactly.

5.2 Heterogeneous and Homogenous Data

As already discussed above, the Sekt ontology annotated corpus contains documents from three different news types, with around 90 documents of each type. In the experiments discussed above, we used two of these parts for training and the third one for testing, resulting in heterogeneous training and test data. In addition, we ran an experiment where two third of the documents from each three parts were put together as training data and the remaining documents were used for testing. Consequently the training and testing data in this case are homogenous.

Using 3-fold cross-validation, the mean results were: conventional $F_1 = 0.850$; distance based $F_1 = 0.932$; Compared to the results of heterogeneous training and testing data listed in Table 4, the homogenous training and testing data helped us achieve better results, but not as much as expected. We attribute this to the fact that the documents in the corpus are annotated with classes from a general ontology (Proton), which does not contain domain specific concepts which only occur in one or two of the subsets.

5.3 Learning curves

In order to establish the effect of data sparseness on the system's performance, we carried out experiments to compare the learning curves for the top 7 classes (listed in Table 2) with those for all classes in the Sekt ontology annotated corpus. The top 7 classes are high level concepts in the Proton ontology and have sub-concepts. For example, *Organisation* subsumes *CommercialOrganisation*, *EducationOrganisation* and *PoliticalEntity*, etc. Figure 2 shows the learning curves, using standard IE micro-averaged F_1 on, respectively, the top 7 classes and all classes. Each experiment constituted of ten runs and in each run N documents were selected at random from the Sekt ontology annotated corpus[9] as training data and all other documents for testing. The results for each experiment are the means over the ten runs, with the confidence intervals at the 95% level.

The results show, unsurprisingly, that the more training documents are used, the better the results are. Secondly, on small training sets (10 or 20 documents), the results for the 7 classes are much better than for all classes. This is due to data sparseness, as each document contains many more examples when the class labels are generalised to the top 7. The training time for 7 classes is also much less than that for all classes, due to the need for training fewer classifiers. Hence, if detailed information is not required, then learning the high level concepts is better, as it needs less training data and is also much faster.

In addition, it is also interesting to compare the different performance measures as well as the different learning algorithms on small training sets, because in many applications there is often only a small amount of annotated training data. Figure 3 presents the results of Hieron, SVM and PAUM for all classes on the Sekt ontology annotated corpus, measured both with conventional and distance-based F_1.

Firstly, for Hieron the difference between the conventional measure and the ontology induced measure is much bigger on small training data than on a larger number of documents. On the other hand, for SVM (and PAUM) the differences between the different measures changed slowly with respect to the size of the dataset. This reflects the different learning mechanisms adopted by Hieron and SVM, respectively. While the only aim of SVM is to classify every instance correctly, the Hieron classifier at first tries to classify an instance as correctly as it possibly can, and if it cannot classify the instance correctly, it then tries to classify it at the lowest possible cost, relative to the correct concept. Therefore, given a sufficiently large training set, Hieron can learn a good classifier for each class and when the learned model is applied it would classify many instance correctly and minimise the cost on a small number of incorrectly classified ones. This leads to a small difference between the conventional measure which considers classes separately and the ontology-based measure which takes into account the relations between classes.

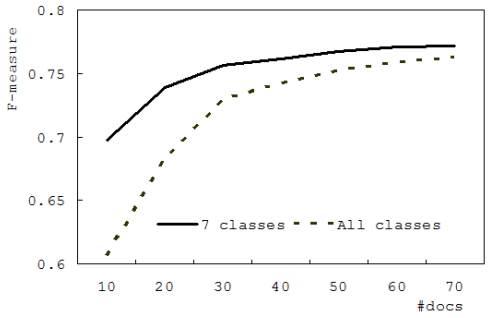

Figure 2: Hieron performance on different amounts of training data for the top 7 classes and all classes respectively

In contrast, if using a small training set, Hieron might not be able to learn a good classifier for some individual classes due to data sparsity. So when this model is applied, it would classify some examples correctly and minimise the cost on many of the misclassified instances. Consequently the ontology-based measure is much higher than the conventional one for small training sets.

Secondly, Hieron has significantly better results than SVM on almost every training set. The ontology-based measures for Hieron are much higher those for SVM, which in turn performs better than PAUM. On the other hand, PAUM is much faster for training and testing than both the SVM and Hieron. The SVM took much longer time than Hieron, in particular for testing. Hence, overall we can conclude that Hieron is a good learning algorithm for OBIE, because it balances performance and computational complexity.

6. EVALUATION AGAINST OTHER METHODS

Since are no existing publically available corpora[10], annotated with more than 10 classes, we evaluated our Hieron-based system on the recently created Sekt corpus. Unfortunately, this makes it difficult to compare our approach to previous systems, as this corpus is new and no previous evaluations have been reported on it, except those reported here.

[9]Each run we selected the same number of documents from each of the three parts of the corpus for training. In other words, the training and test data was homogenous.

[10]The most recent Pascal evaluation challenge for IE systems has not yet released its human-annorated test data, so cannot be used to compare against other systems.

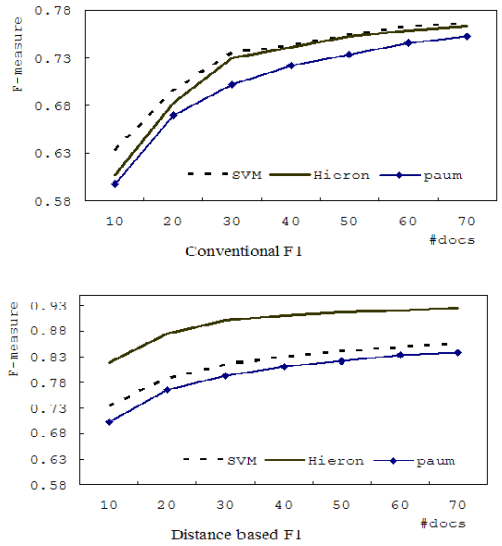

Figure 3: Comparison of Hieron, SVM and PAUM performance on small training sets

On the other hand, there exist several benchmarking corpora on which many IE systems have been evaluated. In order to make a comparison of our Hieron-based learning algorithm to other state of the art systems for IE, we applied our algorithm to the CMU Seminars corpus, which standard benchmarking data for conventional IE.

The Seminars corpus has been used for evaluating many IE systems. This includes rule learning based system such as Rapier [1], BWI [14], SNoW [23] and $(LP)^2$ [6], as well as statistical learning systems such as HMM [12] and maximum entropy (MaxEnt) [3].

The Seminars corpus contains 485 Seminar announcement posts and is annotated with four types of information entities, namely start time (stime), end time (etime), speaker and location of one seminar.

In order to make a fair comparison with other state-of-the-art learning algorithms for IE, our experiment settings followed the methodology of Rapier and $(LP)^2$. First, the results are the average over 10 runs and in each run 243 documents are randomly selected from the corpus as training data and the remaining 242 are used for testing.

Secondly, as far as possible, we used the same features as the other systems used to enable a more informative comparison. In particular, the results discussed here, including our system, do not use any gazetteer information and named entity recogniser output. The only features used in this case are words, capitalisation information, token types, lemmas, and POS tags. Finally, we use the exact match for evaluating the results and do not consider partially correct annotations (see Section 5.1), because the results of the other systems only considered the exact match as well. In other words, an entity is regarded as being identified correctly only if both its start and end tokens are recognised correctly.

Since Hieron needs relations among the entity types, in order to apply it to this corpus, we constructed a trivial ontology which consists of one concept *Thing* as a root node, *Entity* as the only child of *Thing*, and the four entity types *Stime*, *Etime*, *Speaker* and *Location* as the child concepts of *Entity*.

Table 8: Comparison of our Hieron-based system to other learning algorithms on the seminar corpus: F_1 (%) on each slot and macro-averaged F_1. The 95% confidence interval for MA F_1 is presented for Hieron. The best results for each slot and for overall performance appear in bold.

	speaker	location	stime	etime	MA F_1
Hieron	75.5	83.1	95.4	93.7	**87.0**±1.1
SVM	69.0	81.3	94.8	92.7	84.5
$(LP)^2$	**77.6**	75.0	99.0	95.5	86.8
SNoW	73.8	75.2	**99.6**	**96.3**	86.2
MaxEnt	65.3	82.3	**99.6**	94.5	85.4
BWI	67.7	76.7	**99.6**	93.9	84.6
HMM	71.1	**83.9**	99.1	59.5	78.4
Rapier	53.1	73.4	95.9	94.6	79.1

Table 8 presents the results of our Hieron-based algorithm compared to other IE systems on the Seminars corpus. The SVM results are from our earlier experiments with uneven margins SVM on this corpus [18]. The results of the other systems were obtained from the papers cited at the start of this section.

The results show that on each of the four slots our system's performance is not far from the best per-slot measures which are achieved by different systems. As a result, our Hieron-based system obtained the best overall performance, although its performance is not significantly better from the next two systems: $(LP)^2$ and SNoW. From this, we can conclude that our algorithm is also a very good, state-of-the-art learning algorithm for information extraction.

In our view, the excellent perfomance of our Hieron-based system is due to its exploiting the trivial seminar ontology which we created to meet the algorithm's requirements. In this ontology, the four classes of interest have the same distance to each other, whereas the distance is larger between each of them and the non-entity concept, which we added to the ontology as a child concept of the top concept *Thing*. Consequently, due to Hieron's learning mechanism, the classifier for the non-entity class is more different than the classifiers for these 4 target classes. This leads to Hieron making fewer misclassifications than the other learning algorithms (e.g., SVM), which do not explore the asymmetric similarities between the classes and non-entities in the Seminar corpus.

7. CONCLUSION

This paper investigated the adaptation and evaluation on ontology-based information extraction of a large margin hierarchical classification, Perceptron-like algorithm Hieron. The algorithm takes into account the relations among concepts, thus benefiting directly from the ontology structure. We made several modifications to the original Hieron algorithm presented in [9], which, as proven by our evaluation, led to an improved performance.

The algorithm's performance is evaluated on the biggest available semantically annotated corpus, covering 146 concepts from the Proton ontology. Our system is compared to SVM and Perceptron, which are two state-of-the-art learning algorithms for IE. The results showed that our hierar-

chical classification algorithm obtained significantly better results than Perceptron and SVM.

The approach was evaluated on the Sekt ontology-annotated news corpus, as it is the only corpus for the OBIE which is annotated with more than 20 classes from a non-trivial ontology. The problem is that no other systems have been evaluated on it, apart from the SVM and Perceptron comparisons reported here. Unfortunately other recent corpora for IE evaluation use a flat set of labels, thus making them inappropriate for OBIE. However, in order to enable some comparison of our approach to other work, we also ran Hieron on the Seminar corpus, despite it being a flat set of labels. The experimental results showed that our system outperformed again other state-of-the-art learning algorithms, proving that Hieron is well suited for information extraction tasks.

Further work will focus on integrating active learning with the Hieron algorithm and further evaluation on related tasks, e.g., opinion mining.

8. REFERENCES

[1] M. E. Califf. *Relational Learning Techniques for Natural Language Information Extraction*. PhD thesis, University of Texas at Austin, 1998.

[2] N. Cesa-Bianchi, C. Gentile, A. Tironi, and L. Zaniboni. Incremental Algorithms for Hierarchical Classification. In *Neural Information Processing Systems*, 2004.

[3] H. L. Chieu and H. T. Ng. A Maximum Entropy Approach to Information Extraction from Semi-Structured and Free Text. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pages 786–791, 2002.

[4] N. Chinchor. Muc-4 evaluation metrics. In *Proceedings of the Fourth Message Understanding Conference*, pages 22–29, 1992.

[5] P. Cimiano, S. Handschuh, and S. Staab. Towards the Self-Annotating Web. In *Proceedings of WWW'04*, 2004.

[6] F. Ciravegna. $(LP)^2$, an Adaptive Algorithm for Information Extraction from Web-related Texts. In *Proceedings of the IJCAI-2001 Workshop on Adaptive Text Extraction and Mining*, Seattle, 2001.

[7] F. Ciravegna and Y. Wilks. Designing Adaptive Information Extraction for the Semantic Web in Amilcare. In S. Handschuh and S. Staab, editors, *Annotation for the Semantic Web*. IOS Press, Amsterdam, 2003.

[8] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, 2002.

[9] O. Dekel, J. Keshet, and Y. Singer. Large Margin Hierarchical Classification. In *Proceedings of the 21st International Conference on Machine Learning (ICML-2004)*, Canada, 2004.

[10] S. Dill, J. A. Tomlin, J. Y. Zien, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, and A. Tomkins. SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. In *Proceedings of the* 12^{th} *International Conference on World Wide Web (WWW2003)*, pages 178–186, Budapest, Hungary, May 2003.

[11] J. Domingue, M. Dzbor, and E. Motta. Magpie: Supporting Browsing and Navigation on the Semantic Web. In N. Nunes and C. Rich, editors, *Proceedings ACM Conference on Intelligent User Interfaces (IUI)*, pages 191–197, 2004.

[12] D. Freigtag and A. K. McCallum. Information Extraction with HMMs and Shrinkage. In *Proceesings of Workshop on Machine Learnig for Information Extraction*, pages 31–36, 1999.

[13] D. Freitag. Machine Learning for Information Extraction in Informal Domains. *Machine Learning*, 39(2/3):169–202, 2000.

[14] D. Freitag and N. Kushmerick. Boosted Wrapper Induction. In *Proceedings of AAAI 2000*, 2000.

[15] S. Handschuh, S. Staab, and F. Ciravegna. S-CREAM — Semi-automatic CREAtion of Metadata. In *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, pages 358–372, Siguenza, Spain, 2002.

[16] A. Kiryakov, B. Popov, D. Ognyanoff, D. Manov, A. Kirilov, and M. Goranov. Semantic annotation, indexing and retrieval. *Journal of Web Semantics, ISWC 2003 Special Issue*, 1(2):671–680, 2004.

[17] P. Kogut and W. Holmes. AeroDAML: Applying Information Extraction to Generate DAML Annotations from Web Pages. In *First International Conference on Knowledge Capture (K-CAP 2001), Workshop on Knowledge Markup and Semantic Annotation*, Victoria, B.C., 2001.

[18] Y. Li, K. Bontcheva, and H. Cunningham. Using Uneven Margins SVM and Perceptron for Information Extraction. In *Proceedings of Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, 2005.

[19] Y. Li, H. Zaragoza, R. Herbrich, J. Shawe-Taylor, and J. Kandola. The Perceptron Algorithm with Uneven Margins. In *Proceedings of the 9th International Conference on Machine Learning (ICML-2002)*, pages 379–386, 2002.

[20] L. K. McDowell and M. Cafarella. Ontology-Driven Information Extraction with OntoSyphon. In *5th Internal Semantic Web Conference (ISWC'06)*. Springer, 2006.

[21] E. Motta, M. Vargas-Vera, J. Domingue, M. Lanzoni, A. Stutt, and F. Ciravegna. MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup. In *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, pages 379–391, Siguenza, Spain, 2002.

[22] J. Perna and A. Spector. Introduction to the Special Issue on Unstructured Information Management. *IBM Systems Journal*, 43(3), 2004.

[23] D. Roth and W. T. Yih. Relational Learning via Propositional Algorithms: An Information Extraction Case Study. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1257–1263, 2001.