D2.5.2 Report: Quantitative Evaluation Tools and Corpora: version 2

Yaoyong Li, Kalina Bontcheva, Niraj Aswani, Wim Peters, Hamish Cunningham
(University of Sheffield)

Abstract.
EU-IST Integrated Project (IP) IST-2003-506826 SEKT
Deliverable D2.5.2 (WP2.5)

The first part of this deliverable presents the Pascal Challenge on evaluating machine learning methods for Information Extraction, the systems that we entered into the challenge and the evaluation results.
The second part of deliverable focuses on quantitative evaluation of Ontology-Based Information Extraction (OBIE) and uses the semantically annotated corpus, produced in D2.5.1, in order to evaluate the performance of the system.
Keyword list: evaluation, ontology-based information extraction, language processing

Document Id.	SEKT/2005/D2.5.2/v1.0
Project	SEKT EU-IST-2003-506826
Date	31 December 2005
Distribution	public

SEKT Consortium

This document is part of a research project partially funded by the IST Programme of the Commission of the European Communities as project number IST-2003-506826.

British Telecommunications plc.
Orion 5/12, Adastral Park
Ipswich IP5 3RE
UK
Tel: +44 1473 609583, Fax: +44 1473 609832
Contact person: John Davies
E-mail: john.nj.davies@bt.com

Jozef Stefan Institute
Jamova 39
1000 Ljubljana
Slovenia
Tel: +386 1 4773 778, Fax: +386 1 4251 038
Contact person: Marko Grobelnik
E-mail: marko.grobelnik@ijs.si

University of Sheffield
Department of Computer Science
Regent Court, 211 Portobello St.
Sheffield S1 4DP
UK
Tel: +44 114 222 1891, Fax: +44 114 222 1810
Contact person: Hamish Cunningham
E-mail: hamish@dcs.shef.ac.uk

Intelligent Software Components S.A.
Pedro de Valdivia, 10
28006 Madrid
Spain
Tel: +34 913 349 797, Fax: +49 34 913 349 799
Contact person: Richard Benjamins
E-mail: rbenjamins@isoco.com

Ontoprise GmbH
Amalienbadstr. 36
76227 Karlsruhe
Germany
Tel: +49 721 50980912, Fax: +49 721 50980911
Contact person: Hans-Peter Schnurr
E-mail: schnurr@ontoprise.de

Vrije Universiteit Amsterdam (VUA)
Department of Computer Sciences
De Boelelaan 1081a
1081 HV Amsterdam
The Netherlands
Tel: +31 20 444 7731, Fax: +31 84 221 4294
Contact person: Frank van Harmelen
E-mail: frank.van.harmelen@cs.vu.nl

Empolis GmbH
Europaallee 10
67657 Kaiserslautern
Germany
Tel: +49 631 303 5540, Fax: +49 631 303 5507
Contact person: Ralph Traphöner
E-mail: ralph.traphoener@empolis.com

University of Karlsruhe, Institute AIFB
Englerstr. 28
D-76128 Karlsruhe
Germany
Tel: +49 721 608 6592, Fax: +49 721 608 6580
Contact person: York Sure
E-mail: sure@aifb.uni-karlsruhe.de

University of Innsbruck
Institute of Computer Science
Technikerstraße 13
6020 Innsbruck
Austria
Tel: +43 512 507 6475, Fax: +43 512 507 9872
Contact person: Jos de Bruijn
E-mail: jos.de-bruijn@deri.ie

Kea-pro GmbH
Tal
6464 Springen
Switzerland
Tel: +41 41 879 00, Fax: 41 41 879 00 13
Contact person: Tom Bösser
E-mail: tb@keapro.net

Sirma AI EAD, Ontotext Lab
135 Tsarigradsko Shose
Sofia 1784
Bulgaria
Tel: +359 2 9768 303, Fax: +359 2 9768 311
Contact person: Atanas Kiryakov
E-mail: naso@sirma.bg

Universitat Autonoma de Barcelona
Edifici B, Campus de la UAB
08193 Bellaterra (Cerdanyola del Vallès)
Barcelona
Spain
Tel: +34 93 581 22 35, Fax: +34 93 581 29 88
Contact person: Pompeu Casanovas Romeu
E-mail: pompeu.casanovas@uab.es

Executive Summary

The first part of this deliverable presents the Pascal Challenge on evaluating machine learning methods for Information Extraction, the systems that we entered into the challenge and the evaluation results.

The second part of deliverable focuses on quantitative evaluation of Ontology-Based Information Extraction (OBIE) and uses the semantically annotated corpus, produced in D2.5.1, in order to evaluate the performance of the system.

The learning algorithm evaluated here was originally designed for hierarchical classification in [DKS04], which took in account the relations among class labels for a multi-class classification problem. We convert the OBIE task into two multi-class classification problems and then apply the algorithm to them respectively. We also make some modifications on the original algorithm in order to make it more effective.

Information Extraction systems usually compute measures, such as *Precision*, *Recall* and F_1, for each category independent of other categories and then use a measure averaged over the performances for all categories as an overall performance measure. However, these kinds of measures cannot reflect the hierarchical relations of ontologies and therefore an OBIE system requires performance measures which are sensitive to the structure of the given ontology. Therefore, we generalise the commonly used measures, *Precision*, *Recall* and F_1 to OBIE by taking into account concept structure of the ontology.

Contents

1 Introduction **2**

2 The Pascal Challenge on Evaluating Machine Learning for Information Extraction **3**
2.1 Feature selection . . . 3
2.2 The IE Algorithms . . . 4
2.3 Results . . . 5

3 Exploiting the Hierarchical Structure of the Ontology for OBIE **7**
3.1 Ontology-induced performance measures . . . 7
3.2 Large Margin Learning Algorithm Hieron . . . 9
3.3 Applying Hieron to OBIE . . . 12

4 Experimental Datasets **13**

5 Experimental Results **15**
5.1 Flat classification . . . 15
5.2 The Hieron for OBIE . . . 17

6 Conclusion **19**

A Updates on the Ontology Annotated Corpus **22**
A.1 Proton Classes . . . 23

1 Introduction

Information extraction (IE) is a process of automatic extraction of information about pre-defined types of events, entities and relationships from text such as newswire articles and web pages. Ontology based information extraction (OBIE) is a special type of IE, which aims to automatically extract from text instances of concepts in a given ontology. As domain knowledge can be represented by ontology, OBIE is an important approach to extract domain knowledge from unstructured textual sources.

An OBIE system can be build upon hand-crafted rules and knowledge, which require expertise in both domain knowledge and linguistics [MYKK05]. Alternatively, such a system can be built through machine learning approaches, which is the method this paper concentrates on. In comparison to hand-crafted OBIE systems, machine learning ones typically require only some text annotated with concepts as training examples, which are relatively easy to obtain.

Machine learning methods for general IE can be applied to OBIE as well. However, note that the main difference between OBIE and general IE is that the concepts in OBIE have some relations while general IE assumes no specific relation among flat set of labels being extracted. Therefore, in order to build an OBIE system with good performance, we are much more interested in learning algorithms which can exploit rather than ignore the structure of the ontology, especially the subsumption hierarchy.

This deliverable evaluates a large margin Perceptron-like learning algorithm for OBIE. The algorithm was originally designed for hierarchical classification in [DKS04], which took in account the relations among class labels for a multi-class classification problem. We convert the OBIE task into two multi-class classification problems and then apply the Hieron to them respectively. We also make some modifications on the original Hieron to make the algorithm more effective.

This deliverable focuses on quantitative evaluation of OBIE and uses the semantically annotated corpus, produced in D2.5.1, in order to evaluate the performance of Ontology-Based Information Extraction (OBIE).

In order to carry out quantitative evaluation, an ontology-based evaluation metric is required. As concepts in ontology are related to each other in a subsumption hierarchy, the cost (or loss) for an instance of one concept *A* wrongly classified as belonging to another concept *B* may be dependent upon the two particular concepts, which is denoted as $c(A, B)$. Provided some kind of cost for each pair of concepts in a given ontology, if on OBIE system cannot identify an instance of one concept correctly, we would like the system to classify it as one instance of another concept with a smaller cost rather than bigger one (e.g., to classify it as a super-class of the correct class).

IE systems usually compute measures, such as *Precision*, *Recall* and F_1, for each category independent of other categories and then use a measure averaged over the performances for all categories as an overall performance measure. However, these kinds of measures cannot reflect the hierarchical relations of ontologies and therefore an OBIE system requires performance measures which are sensitive to the structure of the given ontology. Therefore, we generalise the commonly used measures, *Precision*, *Recall* and F_1 to OBIE by taking into account concept structure of the ontology.

2 The Pascal Challenge on Evaluating Machine Learning for Information Extraction

The Pascal challenge – evaluating machine learning for information extraction (IE) – aimed at assessing machine learning algorithms for IE from text. The corpus provided consisted of 1100 conference workshop call for papers (CFP), of which 600 were annotated. The annotation covered eleven categories of information entities such as workshop and conference names and acronyms, workshop date, location and homepage. The main purpose of the challenge was to evaluate machine learning algorithms based on the same linguistic features. The only compulsory task is task1, which used 400 annotated documents for training and other 200 annotated documents for testing. See [IC05] for a short overview of the challenge.

The learning methods explored by the participating systems included LP^2, HMM, CRF, SVM, and a variety of combinations of different learning algorithms.

We submitted three systems for task1, task2a and task2b, respectively. As system1 was a combination of system2 and system3, we will describe first system2 and system3 and then introduce system1 (also see [LBC04] for a description of system2). System2 and system3 employed the same framework for applying machine learning to IE — transferring the recognition of information entities into binary classification problems. They also shared the same preprocessing and post-processing procedures. The only difference between system2 and system3 was in the classifiers they used. The SVM with uneven margins was used in system2, while the Perceptron with uneven margins was used in system3 (for details see below).

2.1 Feature selection

The aim of the preprocessing is to form feature vectors from the documents as input to the algorithms. As we iterated through the tokens in each document (including word, punctuation and other symbols) to see if the current token belonged to an information entity or not, we formed a feature vector for each token. The NLP features we used were extracted from the GATE processed documents, as supplied in the corpus, and included token form, word case information, simple categorisation information of each token, and some general entity types from the named entity recognition system ANNIE (e.g., person names, locations, dates, organisations). However, we did not use the POS information provided. The feature vector for a token included the NLP features of all the tokens in a window centered on the current token. The window size (namely the number of words in either side of the current word) was set to 10 in our experiments.

We converted recognition of every type of information entity into two binary classification problems – one was used for deciding whether a token was the start token of the entity and another was for the end token.

2.2 The IE Algorithms

The classification problem derived from IE usually has imbalanced training data, in which positive training examples are vastly outnumbered by negative ones. This is particularly true for smaller data sets where often there are hundreds of negative training examples and only few positive ones. Two approaches have been studied so far to deal with imbalanced data in IE. One approach is to under-sample majority class or over-sample minority class in order to obtain a relatively balanced training data [ZM03]. However, under-sampling can potentially remove certain important examples, and over-sampling can lead to over-fitting and a larger training set. Another approach is to divide the problem into several sub-problems in two layers, each of which has less imbalanced training set than the original one [CMP03, SD03]. The output of the classifier in the first layer is used as the input to the classifiers in the second layer. As a result, this approach needs more classifiers than the original problem. Moreover, the classification errors in the first layer will affect the performance of the second one.

In this deliverable we explore another approach to handle the imbalanced data in IE, namely, adapting the learning algorithms for balanced classification to imbalanced data. We particularly study two popular classification algorithms in IE, Support Vector Machines (SVM) and Perceptron.

[LST03] introduced an uneven margins parameter into the SVM to deal with imbalanced classification problems. They showed that the SVM with uneven margins outperformed the standard SVM on document classification problem with imbalanced training data. Formally, given a training set $\mathbf{Z} = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m))$,where $\mathbf{x}_i$ is the n-dimensional input vector and y_i ($= +1$ or -1) its label, the SVM with uneven margins is obtained by solving the quadratic optimisation problem:

$$\min_{\mathbf{w},\ b,\ \xi} \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{m} \xi_i$$

$$\begin{aligned} \text{s.t.} \quad & \langle \mathbf{w}, \mathbf{x}_i \rangle + \xi_i + b \geq 1 \quad \text{if } y_i = +1 \\ & \langle \mathbf{w}, \mathbf{x}_i \rangle - \xi_i + b \leq -\tau \quad \text{if } y_i = -1 \\ & \xi_i \geq 0 \qquad \text{for } i = 1, ..., m \end{aligned}$$

We can see that the uneven margins parameter τ was added to the constraints of the optimisation problem. τ is the ratio of negative margin to the positive margin of the classifier and is equal to 1 in the standard SVM. For an imbalanced dataset with a few positive examples and many negative ones, it would be beneficial to use larger margin for positive examples than for the negative ones. [LST03] also showed that the solution of the above problem could be obtained by solving a related standard SVM problem by, for example, using a publicly available SVM package[1].

Perceptron is an on-line learning algorithm for linear classification. It checks the training examples one by one by predicting their labels. If the prediction is correct, the example is passed; otherwise, the example is used to correct the model. The algorithm stops when the model classifies all training examples correctly. The margin Perceptron not only classifies every training example correctly

[1]The SVM^{light} package version 3.5, available from http://svmlight.joachims.org/, was used to learn the SVM classifiers in our experiments.

but also outputs for every training example a value (before thresholding) larger than a predefined parameter (margin). The margin Perceptron has better generalisation capability than the standard Perceptron. [LZH+02] proposed the Perceptron algorithm with uneven margins (PAUM) by introducing two margin parameters τ_+ and τ_- into the updating rules for the positive and negative examples, respectively. Similar to the uneven margins parameter in SVM, two margin parameters allow the PAUM to handle imbalanced datasets better than both the standard Perceptron and the margin Perceptron. Additionally, it is known that the Perceptron learning will stop after limited loops only on a linearly separable training set. Hence, a regularisation parameter λ is used in PAUM to guarantee that the algorithm would stop for any training dataset after some updates. PAUM is simple and fast and performed very well on document classification, in particularly on imbalanced training data.

Our experiments showed that the PAUM-based system3 was about 12 times faster than system2 (for instance, 2.17 hours vs 25.85 for the 4-fold cross-validation on the training set for task1).

After classification we obtained the start and end tags of the entities. Then we needed some post-processing procedure to guarantee the consistency of the tags and to try to improve the tags by exploring other information. The procedure we used has three stages. First, in order to guarantee the consistency of the recognition results, a document was scanned from the first to the last token to remove a start tag if there is no end tag immediately following it and remove an end tag without a start tag immediately preceding to it. The second stage filtered out the candidate entity from the output of the first stage using the information about the length of entities obtained from the training set. The third stage put together all possible tags for a piece of text and chose the best one according to the probability which was computed from the output of the classifier (before thresholding) via a Sigmoid function.

Note that system2 and system3 were common in some respects but were also complementary in others: quadratic kernel vs linear kernel and batch optimistation vs on-line optimisation. We therefore implemented system1 as a simple combination of the results from system2 and system3. In other words, the results of system1 were obtained by putting together the tags from system2 and system3 and adopting the results of system2 wherever there was any conflict between the two.

Task2b required the participating system to actively select some training examples from a pool of unannotated documents. We adopted the Gram-Schmidt orthogonalisation algorithm for the selection. The algorithm was successfully used for choosing the negative examples given a few positive examples for the adaptive document filtering task of TREC-2002 (see [CCBC+03]). The Gram-Schmidt algorithm was basically to determine a subset of examples with a pre-defined size, which were furthest from each other and were also furthest from another pre-defined subset (if we have one) in the feature space. See [CSTL02] for more detail about the algorithm.

We did not apply our systems to task3 which allows a system using a richer set of information sources provided by the 500 enrich unannotated documents.

2.3 Results

Firstly, the system of the challenge organisers obtained the best result for Task1, followed by one of our participating systems which combined the uneven margins SVM and PAUM (see [IC05]).

Our SVM and PAUM systems on their own were respectively in the fourth and fifth position among the 20 participating systems.

Secondly, at least six other participating system were also based on SVM but used different IE framework and possibly different SVM models from our SVM system. Our SVM system achieved better results than all those SVM-based systems, showing that the SVM models and the IE framework of our system were quite suitable to IE task.

Thirdly, our PAUM based system was not as good as our SVM system but was still better than the other SVM based systems. The computation time of the PAUM system was about 1/5 of that of our SVM system. Table 1 presents the per slot results and overall performance of our SVM and PAUM systems as well as the system with the best overall result. Compared to the best system, our SVM system performed better on two slots and had similar results on many of other slots. The best system had extremely good results on the two slots, C-acronym and C-homepage. Actually, the F_1 values of the best system on the two slots were more than double of those of every other participating system.

Table 1: Results of our SVM and PAUM systems on CFP corpus: F-measures(%) on individual entity type and the overall figures, together with the system with the highest overall score. The highest score on each slot appears in bold.

SLOT	PAUM	SVM	Best one
W-name	51.9	**54.2**	35.2
W-acronym	50.4	60.0	**86.5**
W-date	67.0	69.0	**69.4**
W-homepage	69.6	70.5	**72.1**
W-location	60.0	**66.0**	48.8
W-submission	70.2	69.6	**86.4**
W-notification	76.1	85.6	**88.9**
W-camera-ready	71.5	74.7	**87.0**
C-name	43.2	47.7	**55.1**
C-acronym	38.8	38.7	**90.5**
C-homepage	7.1	11.6	**39.3**
Micro-average	61.1	64.3	**73.4**

Finally, our systems only used the GATE-processed training and test documents produced by the organiser, not using any external resource. However, internally we carried out a small experiment with using extra linguistic information on task1, which showed improved results in comparison to the more limited NLP features provided in the pascal corpus (evaluated using the muc scorer configuration as supplied by the challenge organisers). The extra information included sentence boundaries, lemma, gazetteers, and a richer entity set (e.g., URL, email). All extra features were provided by the same GATE components, as those used to produce the NLP features in the pascal corpus, but for some reason were not included by the organisers. Although further, more detailed investigation is required, we think that richer linguistic information will be useful for obtaining better performance.

3 Exploiting the Hierarchical Structure of the Ontology for OBIE

The categories of information entities in conventional IE or named entity recognition have no specific relation among them. They are independent of each other. Hence these categories can be learned and recognised independently.

In contrast, as concepts in ontology are related to each other (at the very least through the subsumption hierarchy), it would be beneficial to exploit the hierarchical structure in OBIE.

This paper exploits two aspects of label structure for OBIE. The first aspect is to investigate ontology induced measures for OBIE, which would be used in the learning algorithm. The second one is to investigate a Perceptron based learning algorithm which has a mechanism to effectively handle the structure of concepts in ontology.

3.1 Ontology-induced performance measures

As concepts in ontology are related to each other in a subsumption hierarchy, the cost (or loss) for an instance of one concept *A* wrongly classified as belonging to another concept *B* may be dependent upon the two particular concepts, which is denoted as $c(A, B)$. Provided some kind of cost for each pair of concepts in a given ontology, if on OBIE system cannot identify an instance of one concept correctly, we would like the system to classify it as one instance of another concept with a smaller cost rather than bigger one (e.g., to classify it as a super-class of the correct class).

IE systems usually compute measures, such as *Precision*, *Recall* and F_1, for each category independent of other categories and then use a measure averaged over the performances for all categories as an overall performance measure. However, these kinds of measures cannot reflect the hierarchical relations of ontology and therefore an OBIE system requires performance measures which are sensitive to the structure of the given ontology. Therefore, next we generalise the commonly used measures, *Precision*, *Recall* and F_1 to OBIE by taking into account concept structure of ontology.

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

- n — number of entities in the corpus identified correctly or incorrectly by the OBIE system.
- $n_{missing}$ — number of entities in the corpus which were not recognised 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.

For each pair of concepts *X* and *Y* we assign a cost measure $c(X, Y)$, which is a non-negative number and measures the cost of misclassifying an instance of concept *X* as that of concept *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 $X = Y$.

Using the cost-based error, we define an overall accuracy of the n entities identified by the system as follows:

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

where $e_{cost}(A_i, B_i)$ is the cost of misclassifying the ith instance as class B_i, instead of its correct class A_i.

Using the overall accuracy a_{cost} we can define ontology induced precision and recall, respectively,

$$P_o = \frac{a_{cost}}{n + n_{spurious}}, \qquad R_o = \frac{a_{cost}}{n + n_{missing}}$$

Then, as with the f-measure in "traditional" IE systems, the ontology induced F_1 is defined as the harmonic mean of ontology induced precision and recall:

$$F_{o1} = \frac{2 * P_o * R_o}{P_o + R_o} \tag{2}$$

Note that ontology induced F_{o1} is a generalisation of the standard F_1. Actually, if we define the cost $c(X, Y)$ as the binary function

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

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

In a recent study about hierarchical classification where the classification labels are organised in a tree, $c(X, Y)$ was often defined as the distance $\gamma(X, Y)$ of the two nodes X and Y in the tree, e.g. the number of edges in the shortest path connecting nodes X and Y, which was used to define the tree induced error in [DKS04] and several other papers.

[AR96] proposed four criteria for measuring closeness of two concepts organised in a graph:

1. Dependent on length of the shortest path connecting the two concepts involved.
2. The concepts in a deeper part of the hierarchy should be closer.
3. Concepts in a dense part of the hierarchy should be relatively closer than those in sparse region.
4. Independent of number of concepts in the graph.

We believe that a good cost measure for ontology should also be compatible with the above criteria. Unfortunately, the cost measure using distance directly violates the second and third criterion, although it is indeed compatible with the other two.

[MYKK05] proposed a new cost measure BDM which can be used for OBIE. The BDM measure is based on the distance of two nodes in the ontology graph. and satisfies all four criteria.

In detail, given key node K and response node R in an ontology graph, the BDM measure is

$$BDM(K,R) = \frac{BR * CP/n0}{BR * CP/n0 + DPK/n2 + DPR/n3} \tag{4}$$

where CP is the length of the shortest path from the root concept to MSCA node (the most specific concept common to the key and response nodes). DPK and DPR are the lengths of the shortest paths from MSCA to the key and response nodes, respectively. $n2$ and $n3$ are the averaged lengths of chains (from the root node to a leaf node) containing the key and response nodes, respectively. $n0$ is the averaged length of all chains in the ontology graph, which is used in the formula for normalising the two specific chain lengths $n2$ and $n3$ such that the measure is not sensitive to the size of the ontology (refer to the fourth criterion). $n0$, $n2$ and $n3$ are used together for representing the vertical density of the local area containing the key and response nodes. BR is used for measuring the traversal density of the local area, which is computed as the averaged branches of the nodes between the MSCA node and the key node and the nodes between the MSCA node and the response node and is normalised by the averaged number of branches over all nodes in the graph.

Finally we define the DBM measure based cost as $e_{cost}(R,K) = 1 - BDM(R,K)$, as BDM measure is between 0 and 1 and is in proportion to the closeness of two nodes in graph.

3.2 Large Margin Learning Algorithm Hieron

[DKS04] proposed a large margin learning algorithm *Hieron* for hierarchical classification. 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 hierarchical taxonomy. Next we describe the learning algorithm and our modifications over the original Hieron, and in the next subsection discuss how to apply it to OBIE.

For hierarchical classification problem, the Hieron exploits the hierarchical structure of class labels. It learns one model for every class, 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 the class C, then we want the class C to be close to the true class of the example in the hierarchical structure.

Suppose we want to solve 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, we define $\mathcal{P}(v)$ to be the set of labels along the path from root to v inclusive.

We receive a training set $\mathcal{S} = \{(\mathbf{x}_i, y_i) : i = 1, \ldots, m\}$ of instance-label pairs, where each $\mathbf{x}_i \in \mathcal{X}$ and each $y_i \in \mathcal{Y}$. The learning algorithm Hieron 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:

$$f(\mathbf{x}) = \mathrm{argmax}_{v \in \mathcal{Y}} \langle \mathbf{W}^v, \mathbf{x} \rangle \tag{5}$$

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, the Hieron does not deal directly with the set of prototypes but rather with the difference between each prototype 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. We define the difference weight vector as $\mathbf{w}^v = \mathbf{W}^v - \mathbf{W}^{\mathcal{A}(v)}$. Each prototype is now decomposed into the sum

$$\mathbf{W}^v = \sum_{u \in \mathcal{P}(v)} \mathbf{w}^u \tag{6}$$

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

$$\sum_{v \in \mathcal{P}(y_i)} \langle \mathbf{w}^v, \mathbf{x}_i \rangle - \sum_{u \in \mathcal{P}(r)} \langle \mathbf{w}^u, \mathbf{x}_i \rangle \geq \sqrt{\gamma(y_i, r)}, \quad \forall (\mathbf{x}_i, y_i) \in \mathcal{S} \text{ and } \forall r \in \mathcal{Y} \backslash \{y_i\} \tag{7}$$

The difference in (7) is a generalisation of the notion of margin employed by multi-class problems for hierarchical classification (see [DKS04] for details). However, this assumption can be loosened if we introduced some regulation parameter into the learning algorithm, for details see below.

Algorithm 1 Batch Hieron

Require: A training set $\mathcal{S} = \{(\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y} : i = 1, \ldots, m\}$ satisfying the assumptions (7)

Initialise: $\forall v \in \mathcal{Y}$: $\mathbf{w}_0^v = \mathbf{0}$; $t = 0$
repeat
  for each $(\mathbf{x}_i, y_i) \in \mathcal{S}$ **do**
    compute $(\hat{y}_i, l_i) = (\text{argmax}, \max)_{y \in \mathcal{Y}} L(\{\mathbf{w}^v\}, \mathbf{x}_i, y_i, y)$
    where $L(\cdot)$ is the loss function defined in (8)
    if $l_i > 0$ **then**
      update
        $\mathbf{w}_{t+1}^v = \mathbf{w}_t^v + \alpha_i \mathbf{x}_i, \;$ if $v \in \mathcal{P}(y_i) \backslash \mathcal{P}(\hat{y}_i)$
        $\mathbf{w}_{t+1}^v = \mathbf{w}_t^v - \alpha_i \mathbf{x}_i, \;$ if $v \in \mathcal{P}(\hat{y}_i) \backslash \mathcal{P}(y_i)$
      where $\alpha_i = l_i / \left(\gamma(y_i, \hat{y}_i) \|\mathbf{x}_i\|^2\right)$
      $t = t + 1$
    end if
  end for
until no update made within the **for** loop
$\{\mathbf{w}_t^v : v \in \mathcal{Y}\}$

The Hieron learning algorithm is described in Algorithm similar to the Perceptron algorithm but, unlike the Perceptron where only one weight vector is learned, it learns many weight vectors.

The algorithm initialises each of the weight vectors $\{\mathbf{w}^v : v \in \mathcal{Y}\}$ as zero vector and updates a weight vector only if a prototype related with it made 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 (7). 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,

$$L(\{\mathbf{w}^v\}, \mathbf{x}_i, y_i, y) = \sum_{u \in \mathcal{P}(y)} \langle \mathbf{w}^u, \mathbf{x}_i \rangle - \sum_{v \in \mathcal{P}(y_i)} \langle \mathbf{w}^v, \mathbf{x}_i \rangle + \sqrt{\gamma(y_i, y)} \tag{8}$$

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 weight vectors. Otherwise, from all training examples $(\mathbf{x}_i, y_i)$ for which the margin requirement (7) is violated by the current weight vectors, choose the label $\hat{y}_i$ that violate the margin requirement the most, namely it has the maximal value of the function (8), and update the current weight vectors comprising the two prototypes $\mathbf{W}^{y_i}$ and $\mathbf{W}^{\hat{y}_i}$, respectively, as illustrated in the Figure 1.

As we said above, in order to ensure that adjacent vertices in the label tree have similar prototypes, the Hieron needs to keep the norms of weight vector $\mathbf{w}$ as small as possible. By initialising all the weight vectors with zero and only updating them when it is necessary, the algorithm does try to keep the norms of weight vector small.

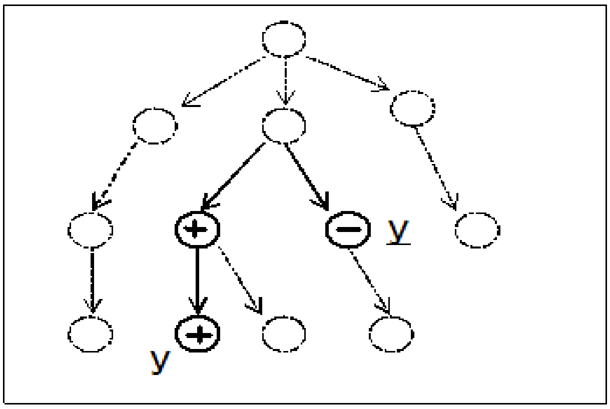

Figure 1: An illustration of the update in Hieron algorithm. When a training example $\mathbf{x}$ with label y is predicted mistakenly as label $\underline{y}$, only the weight vectors associated with the nodes in the shortest path linking nodes y and $\underline{y}$ but except the MSCA node are updated. In other words, only the nodes depicted using solid lines are updated, in which the symbol '+' means inceasing the correspondng weight vector by the example $\mathbf{x}$ and the symbol '-' means decreasing the weight vector by $\mathbf{x}$.

The learning algorithm described above is basically the same as the original Hieron batch learning algorithm presented in [DKS04]. However we have made some modifications in our implementation, which are discussed next:

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

- The Hieron learning algorithm requires that the training set is compatible with the margin conditions described in equation (7). The learning algorithm would stop after a finite number of loops only if the training set satisfies the margin condition. Otherwise, it would run infinitely.

  This might be a problem because we do not know in advance whether or not a training set satisfies the margin condition. However, we can introduce some regulation parameter into the algorithm such that the learning would stop after some loops on any training set. The regulation parameter is similar to that used for Perceptron (see [LZH$^+$02]).

- In [DKS04] two types of learning models were distinguished. One type was the weight vectors obtained at the end of learning, namely $\{\mathbf{w}_t^v : v \in \mathcal{Y}\}$, which corresponds to the standard learning model of Perceptron. Another one 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

  $$\mathbf{w}^v = \frac{1}{m}\sum_{i=1}^{m}\mathbf{w}_i^v \tag{9}$$

  It was showed in [DKS04] that the averaged weight vectors had better results than the last weight vectors in most cases. We will compare the two types of weight vectors in our experiment as well.

3.3 Applying Hieron to OBIE

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 for the Hieron algorithm.

First, we convert the OBIE task into two hierarchical classification problems. As shown in [LBC05a], in order to use classifiers for information extraction, it was efficient to check tokens in text one by one and formalise the task of extracting one type of information entity 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 classifiers are trained – one for recognising the beginning of mentions of the given class and one for the end.

Secondly, for each hierarchical classification problem derived from OBIE, for example for start tokens of a given class, we check tokens one by one to see whether or not they are start tokens of the information entity we are interested in. It is certain that most tokens are not start token for any class (e.g., spaces). Therefore, in order to apply the Hieron to OBIE, we added one node into the ontology as child of the root node, that represents the concept of non-start token (or no-end token). However, this added concept would not be considered when we computed the tree-induced F_1 or other ontology based measures.

Thirdly, note that the Hieron algorithm requires that the classes are organised in a tree. However, for some OBIE tasks, the concept structure in the ontology is not a tree. In fact, in many cases the concepts in ontology are organised in a hierarchy and, if we try to represent the structure by a tree, then some of the concepts may occur in two or more different nodes in the tree. In other words, if we require that one concept is represented only by one node, then some nodes in the tree may have one more parent nodes, as illustrated in Figure 2. The Proton ontology used in our experiments (see below) is one example of this kind of ontology. It is organised in a tree structure. However, some concepts (for instance occur in more than one different places in the structure. For instance, the concept *proton:Announcement* occurs in four different places and *proton:CEO* occurs in two different places in the Proton ontology. In our experiments we adapted the Hieron algorithm to the tree-like structure of the Proton ontology. We did not make any change in the Hieron learning for the tree-like structure, because the learning only involves the shortest path between a pair of nodes which can be obtained unambiguously from the tree-like structure. In the application, the only modification we made was to compute one prototype vector for each path from the root node to the node considered according to the formula (6), rather than only one prototype for a node in the case of tree as there is only one path from root to the node in tree. Then, given one test example, we compared the inner products between the example and every prototype vectors and assign to the example the class one of which prototype is most relevant to the example.

Finally, we replace the distance $\gamma(X, Y)$ in the Hieron with the cost measure $c(X, Y)$ between two concepts in the ontology. Therefore, we can learn classifiers which are optimised according to a particular cost measure we are interested in. In the case of BDM measure, we define the cost $c_{bdm}(X, Y)$ as

$$c_{bdm}(X, Y) = L * (1 - BDM(X, Y))$$

where L is the length of the longest path among the shortest paths linking any two nodes in the graph.

4 Experimental Datasets

The corpus used in our experiments consists of news articles. The articles were divided into three subsets according to article's theme, namely business, international-politics and UK-politics, which has 91, 99 and 100 articles, respectively. The corpus was annotated according to the Proton ontology[2]. The Proton ontology corresponds to a hierarchical structure with 10 levels and the maximal path length is 16. The news corpus was annotated with 169 concepts of the Proton ontology,

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

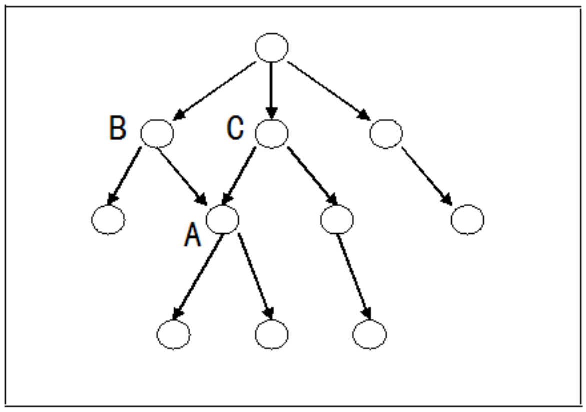

Figure 2: A tree-like structure where the node *A* has two parent nodes *B* and *C*.

which span from the 3rd to 10th level of the hierarchical structure. Hereafter we will refer to the corpus as the sekt ontology-annotated news corpus. Table 2 presents the distribution of concepts with different numbers of mentions in the corpus.

In order to examine the effect of data sparseness on algorithm performance, we also took a set of 8 classes, which are broadly equivalent to labels used in traditional IE systems (e.g., Person, Location, etc). Table 3 presents the numbers of mentions of each of the 8 concepts in each part of the corpus.

Table 2: Distribution of concepts with different numbers of instances in the sekt ontology-annotated news corpus.

#examples of concept	1	2	3	4	5	6 – 10	11 – 20	>20
#concepts	3	12	16	6	3	11	18	100

Table 3: Numbers of instances of the 8 concepts in the three subsets of the sekt ontology-annotated news corpus, respectively.

	#Doc	Person	Loc	Org	Money	Number	Position	Temporal	Time
Business	91	333	593	1446	520	713	32	121	735
Int	99	908	1871	865	88	524	130	110	526
UK	100	844	855	883	207	530	105	107	657

The corpus was pre-processed with the open-source ANNIE system, which is part of GATE [CMBT02]. This enabled us to use a number of linguistic (NLP) features, in addition to information already present in the document such as words and capitalisation information. The NLP 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 recogniser.

Feature vector, as the input to learning algorithm, was derived from the NLP features of each token in the following way:

1. All possible features from the training documents are collected and indexed with a unique identifier, and each dimension of the feature vector corresponds to one feature (e.g. a given token string such as “Time” or a part-of-speech (POS) category such as “CD”).

2. For each token, each component of the feature vector that corresponds to the value of the respective NLP feature are set to 1, and all other components are set to 0.

Since in information extraction the context of the token is usually as important as the token itself, the input vector of the learning algorithm needs to take into account features of the preceding and following tokens, in addition to those of the given token. In our experiments the same number of left and right tokens was taken as a context. In other words, the current token was at the centre of a window of tokens from which the features are extracted. This is called a *window size*. Therefore, for example, when the window size is 3, the algorithm uses features derived from 7 tokens: the 3 preceding, the current, and the 3 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 those of its neighboring tokens.

See [LBC05a] for more detailed description of the feature vector representation used in the experiments.

5 Experimental Results

We have run experiments using the learning algorithm Hieron on the sekt ontology-annotated news corpus. For evaluating the Hieron algorithm on OBIE, we also compare results of the Hieron with those of SVM and the uneven margins Perceptron. In the experiments using SVM and Perceptron, we did flat classification on the sekt ontology-annotated news corpus. Flat classification on a corpus means ignoring the relationships between labels and treating every label separately from other labels. The Hieron algorithm is very similar to the uneven margins Perceptron except that the Hieron takes into account the relationship among labels while the Perceptron treats the label independently.

5.1 Flat classification

Since the news corpus was recently annotated with the ontology, we would like to check the annotation quality before we carry on OBIE experiment on it. Fortunately, the news corpus was also annotated with named entities. Table 4 shows some statistical information about those named entities in the news corpus. We can see that the named entity annotation and the ontology have at least 4 categories in common, namely Person, Location, Organisation and Money. Therefore, we can compare the results for the named entities annotation with those for the sekt ontology-annotated news corpus to check the quality of the annotation.

Table 4: Numbers of named entities in every subset of the News corpus, respectively.

	Person	Location	Organisation	Date	Money	Percent
Business	343	637	1431	790	497	314
Int	1081	2030	858	701	78	86
UK news	897	816	811	635	94	54

Table 5 compares the results on the four common classes of the sekt ontology-annotated news corpus and the named entity news corpus. For each of the two corpus, we used SVM for flat classification. and run three experiments, each of which used one subset of the news corpus as test set and other two subsets as training set. We can see that for the common categories the results with the ontology were significantly worse than those for named entities, showing that the ontology-annotated corpus was harder.

A comparative analysis of the two corpora showed that the difference comes from the positioning of the beginnings and ends of labels in the text. More specifically, the ontology-annotated corpus would annotate with wider spans, often covering the entire phrase, whereas the other corpus would only annotate the names themselves. An example is ”US president George Bush” would be annotated as a class Person in the ontology corpus, whereas only George Bush would be annotated as Person entity in the other case. In addition, in the ontology corpus, US would be annotated as a location, thus requiring the token US to be classified both as a beginning of a Location class and beginning of a Person class. However, our formalisation of the OBIE task supports only 1 classification, either as location or as person. This therefore leads to lower performance figures overall. At present we are working on a version of the ontology-annotated corpus where the boundaries match closer these of the other corpus.

Table 5: Comparison of experimental results between sekt ontology-annotated news corpus and the named entity news corpus: F_1 for each of the four common classes. SVM was used in each experiment as flat classification.

	Person	Location	Organisation	Money
	Ontology corpus			
Business	88.1	82.1	81.4	74.7
Int	82.0	79.6	70.4	78.7
UK	84.7	75.8	70.3	54.7
	Name entity corpus			
Business	90.5	91.0	86.0	93.7
Int	91.1	93.9	85.4	98.1
UK	92.7	93.8	80.5	98.9

5.2 The Hieron for OBIE

The Hieron algorithm exploits the relationships among labels. So we can expect that the Hieron would perform better on OBIE than on flat classification, since OBIE can be seen as a multi-classification problem with structure of labels. Next we compare the Hieron with two popular learning algorithms for IE, the SVM and Perceptron.

In our experiments, we used the uneven margins SVM and the Perceptron with uneven margins, instead of the standard SVM and Perceptron algorithms, because the uneven margins SVM and Perceptron had better performances than the respective standard models for IE (see [LBC05b]). We made comparison on the sekt ontology-annotated news corpus. For both SVM and Perceptron, we apply them to the corpus as solving a general IE problem, taking no consideration of the label structure of the corpus. For the Hieron, as shown in Section 3.2, we took into account the label structure as well as the cost measure $c(X, Y)$ between the two nodes.

Table 6 presents the results of the three learning algorithms on the sekt ontology-annotated news corpus, measured by the conventional micro-averaged F_1 as well as the ontology induced F_1 as shown in (2). For the ontology induced F_1 we used the distance between two nodes as cost. We run three experiments for each algorithm by using each of three subsets of corpus for testing and the other two subsets for training. We can see that the Hieron achieved a significantly higher distance-based F_1 than the SVM and Perceptron. This was mainly due to the optimisation mechanism built into the Hieron for the ontology-induced measure. It was a bit surprising that the Hieron also performed better on two of the three experiments than both the SVM and Perceptron in term of the conventional F_1 which does not consider the relations among the labels at all, showing that considering the relations of labels may also benefit extraction of entities of individual categories.

Table 6: Comparisons of the Hieron with SVM and Perceptron learning on OBIE: micro-averaged F_1 (%) and ontology induced F_1(%) which was based on the distance of labels. PAUM refers to a variant of Perceptron learning, Perceptron Algorithm with uneven margins.

	Micro-averaged F_1			Distance induced F_1		
	PAUM	SVM	Hieron	PAUM	SVM	Hieron
Business	61.7	**65.5**	56.2	65.2	72.7	**75.6**
Int	52.7	58.5	**59.8**	57.2	67.3	**77.1**
UK	52.4	54.4	**59.5**	58.0	63.6	**75.6**

As said in Section 3.2, we have made some modifications on the Hieron algorithm presented in [DKS04]. The original batch Hieron algorithm just ran one cycle on the training set and then used as learning model either the last updated weight vectors or the mean of all the weight vectors obtained in the learning round. In our experiment we allow many 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 for any training examples.

Table 7 presents the results of the original Hieron and the ones with our modifications, using the business and international politics subsets for training and the UK politics subset for testing. We can see that the averaged weight performed better than the last weight, particularly for the original Hieron algorithm, which is compatible with the results in [DKS04]. Multi-loop learning had significantly better results than the single loop learning, showing that multi-loop learning explored more regularity than the single loop could. While the learning algorithm with regulation parameter had similar performance as the multi-loop learning (300 loops) without it, the training time of the former was only about tenth of training time of the latter. Actually with regulation parameter $\lambda = 0.1$ only 28 training loops were run before the learning stopped.

Table 7: Comparisons of the different settings of the Hieron: micro-averaged F_1 (%) and ontology induced F_1 (%) which was based on the distance of nodes. For the original algorithm (single loop) and our modifications (multi-loop and using regulation), we report the results for the last weight as well as the mean of all obtained weights during learning.

	Single loop		Multi-loop		Regulation	
	Last	Mean	Last	Mean	Last	Mean
Micro-averaged F_1	47.9	51.3	59.1	59.5	59.5	**59.7**
Distance induced F_1	69.3	71.3	74.0	74.4	**75.6**	75.5

[DKS04] used the distance between two nodes as the cost in the Hieron learning. However, as discussed in Section 3.1, the BDM measure looks a better measure of closeness between two concepts in ontology than the distance. So we may use the BDM based cost in the Hieron learning as well. Table 8 compares the experimental results of the BDM based cost with those of the distance cost. The BDM based cost had slightly lower results than the distance cost in all the three F_1 measures, the conventional F_1, the distance based F_1 and the BDM based F_1. We thought that the BDM based F_1 could be improved in the experiments using BDM based cost in the Hieron, since the Hieron using BDM based cost was supposed to be optimised with the BDM measure. However, we did not obtain the improved BDM F_1, which need further investigation.

Table 8: Comparison of the BDM based cost with the distance based cost used in the Hieron: conventional micro-averaged F_1 (%), the distance based F_1 (%) and the BDM based F_1 (%). The UK-political subset of the news corpus as test set and other two subsets as training set.

	Conventional F_1	Distance based F_1	BDM based F_1
Distance cost	59.5	75.6	71.7
BDM based cost	59.3	75.2	71.2

6 Conclusion

This deliverable focused on quantitative evaluation of OBIE and used the semantically annotated corpus, produced in D2.5.1, in order to evaluate the performance of Ontology-Based Information Extraction (OBIE).

In particular, we investigated a large margin Perceptron-like algorithm Hieron for OBIE. The algorithm takes into account the relations among the concepts in the ontology. Hence it can exploit the structure of concepts in an ontology. We made several modifications on the original Hieron algorithm presented in [DKS04]. Our experiment results showed that the modifications led to improved performance.

The algorithm's performance is compared to the SVM and Perceptron, two popular learning algorithms for IE. The Hieron obtained better results than Perceptron and SVM in terms of the ontology-induced measure as well as the conventional precision and recall measures for IE.

In order to carry out quantitative evaluation, an ontology-based evaluation metric was required, as traditional IE metrics do not take into account the hierarchical relations in ontologies and therefore we investigated performance measures which are sensitive to the structure of the given ontology. As a result, we generalised the commonly used measures, *Precision*, *Recall* and F_1 to OBIE by taking into account the concept structure of the ontology.

References

[AR96] E. Agirre and G. Rigau. Word sense disambiguation using conceptual density. In *Proc. of 16th International Conference on Computational Linguistics*, volume 1, pages 16–23, Copenhagen, Denmark, 1996.

[CCBC+03] N. Cancedda, N .Cesa-Bianchi, A. Conconi, C. Gentile, C. Goutte, T. Graepel, Y. Li, J.M. Renders, and J .Shawe-Taylor. Kernel methods for document filtering. In E. M. Voorhees and Lori P. Buckland, editors, *Proceedings of The Eleventh Text Retrieval Conference (TREC 2002)*. The NIST, 2003.

[CMBT02] 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.

[CMP03] X. Carreras, L. Màrquez, and L. Padró. Learning a perceptron-based named entity chunker via online recognition feedback. In *Proceedings of CoNLL-2003*, pages 156–159. Edmonton, Canada, 2003.

[CSTL02] N. Cristianini, J. Shawe-Taylor, and H. Lodhi. Latent semantic kernels. *Journal of Intelligent Information System*, 18(2/3):127–152, 2002.

[DKS04] 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.

[IC05] N. Ireson and F. Ciravegna. Pascal Challenge The Evaluation of Machine Learning for Information Extraction. In *Proceedings of Dagstuhl Seminar Machine Learning for the Semantic Web (http://www.smi.ucd.ie/Dagstuhl-MLSW/proceedings/)*, 2005.

[LBC04] Y. Li, K. Bontcheva, and H. Cunningham. An SVM Based Learning Algorithm for Information Extraction. Machine Learning Workshop, Sheffield, 2004. `http://gate.ac.uk/sale/ml-ws04/mlw2004.pdf`.

[LBC05a] Y. Li, K. Bontcheva, and H. Cunningham. SVM Based Learning System For Information Extraction. In *Proceedings of Sheffield Machine Learning Workshop*, Lecture Notes in Computer Science. Springer Verlag, 2005.

[LBC05b] 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.

[LST03] Y. Li and J. Shawe-Taylor. The SVM with Uneven Margins and Chinese Document Categorization. In *Proceedings of The 17th Pacific Asia Conference on Language, Information and Computation (PACLIC17)*, Singapore, Oct. 2003.

[LZH$^+$02] 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.

[MYKK05] D. Maynard, M. Yankova, A. Kourakis, and A. Kokossis. Ontology-based information extraction for market monitoring and technology watch. In *ESWC Workshop "End User Apects of the Semantic Web")*, Heraklion, Crete, 2005.

[SD03] A. De Sitter and W. Daelemans. Information extraction via double classification. In *Proceedings of ECML/PRDD 2003 Workshop on Adaptive Text Extraction and Mining (ATEM 2003)*, Cavtat-Dubrovnik, Croatia, 2003.

[ZM03] J. Zhang and I. Mani. KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction. In *Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Datasets*, 2003.

A Updates on the Ontology Annotated Corpus

The manually annotated onto-news-corpus has annotations of type *Mention*, where each annotation has a feature *class* that contains one of the class values from the *Proton*[3] ontology. For the corpus to contain annotations only over the proper-nouns, some post-processing was required.

We used the ANNIC Tool to identify such annotations. The corpus was processed with the GATE English Tokenizer, Sentence Splitter and Part-of-Speech tagger before it was indexed with the ANNIC tool. The ANNIC Search PR, which given an annotation pattern query returns the relevant annotations in context, was used to identify non-proper-name annotations from the corpus. Some of these annotations were removed manually and for the rest, we use the JAPE grammar. There were three issues which required to be dealt with. These include

1. **removing annotations over the text which cannot be identified as proper nouns** For example:
   - "market" annotated as *Market*
   - "international markets" annotated as *Market*
   - "members" annotated as *Person*
   - "report" annotated as *Document*
   - "regions" annotated as *Location*
   - "subscriber" annotated as *Person*
   - "medical and scientific journals" annotated as *Magazine*
   - "stock market" annotated as *StockExchange*
   - "passengers" annotated as *Person*
   - all annotations annotated as *Webpages*
   - all *Mention* annotations that satisfy the following Part-of-Speech tags pattern $(NN|NNS)(NN|NNS)*(ANY)*$ where $(NN|NNS)$ means the token with the noun (NN) or the plurarity of noun (NNS) part-of-speech tag, $(NN|NNS)*$ means zero or more tokens with NN or NNS part-of-speech tag, and $(ANY)*$ means zero or more tokens with any part-of-speech tag (e.g. health clubs)
2. **fixing the incorrect boundaries of annotations** For example:
   - annotations marked as *Person*
     - "BT's finance director Philip Hampton" corrected to "Philip Hampton"
     - "James Hogan, chief operating officer" corrected to "James Hogan"
     - "internet analyst Mary Meeker" corrected to "Mary Meeker"
   - annotations which do not comply with the underlying token boundaries

[3]http://proton.semanticweb.org

  - "British Telecommunication" corrected to "British Telecommunications"
  - "Square Mile's" corrected to "Square Mile"

3. **modifying the incorrect class values** For example:

  - all annotations marked as *Time* were changed to *TimeInterval*
  - in the text "08.08.01 : 30,000" where "1:30" was annotated as *TimeInterval*. This was removed and two separate annotations were created. 1) "08.08.01" as *Date* and "30,000" as *Number*

A.1 Proton Classes

The corpus has been annotated with the Proton ontology. The table A.1 lists the classes used for annotating the corpus and their relevant URIs in the proton ontology (see http://proton.semanticweb.org).

Classes	**URIs from Proton**
Entity	http://proton.semanticweb.org/2005/04/protons#Entity
Abstract	http://proton.semanticweb.org/2005/04/protont#Abstract
Agent	http://proton.semanticweb.org/2005/04/protont#Agent
Document	http://proton.semanticweb.org/2005/04/protont#Document
Event	http://proton.semanticweb.org/2005/04/protont#Event
GeneralTerm	http://proton.semanticweb.org/2005/04/protont#GeneralTerm
Position	http://proton.semanticweb.org/2005/04/protont#JobPosition
language	http://proton.semanticweb.org/2005/04/protont#Language
Location	http://proton.semanticweb.org/2005/04/protont#Location
Number	http://proton.semanticweb.org/2005/04/protont#Number
BusinessObject	http://proton.semanticweb.org/2005/04/protont#Object
Object	http://proton.semanticweb.org/2005/04/protont#Object
Organization	http://proton.semanticweb.org/2005/04/protont#Organization
Person	http://proton.semanticweb.org/2005/04/protont#Person
Product	http://proton.semanticweb.org/2005/04/protont#Product
Statement	http://proton.semanticweb.org/2005/04/protont#Statement
TimeInterval	http://proton.semanticweb.org/2005/04/protont#TimeInterval
Accident	http://proton.semanticweb.org/2005/04/protonu#Accident
Address	http://proton.semanticweb.org/2005/04/protonu#Address
Airline	http://proton.semanticweb.org/2005/04/protonu#Airline
AirplaneModel	http://proton.semanticweb.org/2005/04/protonu#AirplaneModel
Airport	http://proton.semanticweb.org/2005/04/protonu#Airport
Archipelago	http://proton.semanticweb.org/2005/04/protonu#Archipelago
AstronomicalObject	http://proton.semanticweb.org/2005/04/protonu#AstronomicalObject
Bank	http://proton.semanticweb.org/2005/04/protonu#Bank
Bay	http://proton.semanticweb.org/2005/04/protonu#Bay
Book	http://proton.semanticweb.org/2005/04/protonu#Book
Brand	http://proton.semanticweb.org/2005/04/protonu#Brand

Bridge	http://proton.semanticweb.org/2005/04/protonu#Bridge
Building	http://proton.semanticweb.org/2005/04/protonu#Building
BusinessAbstraction	http://proton.semanticweb.org/2005/04/protonu#BusinessAbstraction
CalendarMonth	http://proton.semanticweb.org/2005/04/protonu#CalendarMonth
CalendarYear	http://proton.semanticweb.org/2005/04/protonu#CalendarYear
Camp	http://proton.semanticweb.org/2005/04/protonu#Camp
Capital	http://proton.semanticweb.org/2005/04/protonu#Capital
CarModel	http://proton.semanticweb.org/2005/04/protonu#CarModel
Chairman	http://proton.semanticweb.org/2005/04/protonu#Chairman
Channel	http://proton.semanticweb.org/2005/04/protonu#Channel
Charity	http://proton.semanticweb.org/2005/04/protonu#Charity
ChemicalCompound	http://proton.semanticweb.org/2005/04/protonu#ChemicalCompound
City	http://proton.semanticweb.org/2005/04/protonu#City
CommercialOrganization	http://proton.semanticweb.org/2005/04/protonu#CommercialOrganization
Company	http://proton.semanticweb.org/2005/04/protonu#Company
Continent	http://proton.semanticweb.org/2005/04/protonu#Continent
Country	http://proton.semanticweb.org/2005/04/protonu#Country
CountryCapital	http://proton.semanticweb.org/2005/04/protonu#CountryCapital
County	http://proton.semanticweb.org/2005/04/protonu#County
Currency	http://proton.semanticweb.org/2005/04/protonu#Currency
Date	http://proton.semanticweb.org/2005/04/protonu#Date
DayOfMonth	http://proton.semanticweb.org/2005/04/protonu#DayOfMonth
DayOfWeek	http://proton.semanticweb.org/2005/04/protonu#DayOfWeek
Desert	http://proton.semanticweb.org/2005/04/protonu#Desert
Disease	http://proton.semanticweb.org/2005/04/protonu#Disease
Drug	http://proton.semanticweb.org/2005/04/protonu#Drug
EducationalOrganization	http://proton.semanticweb.org/2005/04/protonu#EducationalOrganization
EMail	http://proton.semanticweb.org/2005/04/protonu#EMail
Employee	http://proton.semanticweb.org/2005/04/protonu#Employee
Facility	http://proton.semanticweb.org/2005/04/protonu#Facility
Festival	http://proton.semanticweb.org/2005/04/protonu#Festival
Forest	http://proton.semanticweb.org/2005/04/protonu#Forest
GlobalRegion	http://proton.semanticweb.org/2005/04/protonu#GlobalRegion
Government	http://proton.semanticweb.org/2005/04/protonu#Government
GovernmentOrganization	http://proton.semanticweb.org/2005/04/protonu#GovernmentOrganization
Gulf	http://proton.semanticweb.org/2005/04/protonu#Gulf
Harbor	http://proton.semanticweb.org/2005/04/protonu#Harbor
IndustrySector	http://proton.semanticweb.org/2005/04/protonu#IndustrySector
Institute	http://proton.semanticweb.org/2005/04/protonu#Institute
InsuranceCompany	http://proton.semanticweb.org/2005/04/protonu#InsuranceCompany
InternationalOrganization	http://proton.semanticweb.org/2005/04/protonu#InternationalOrganization
Island	http://proton.semanticweb.org/2005/04/protonu#Island
LandRegion	http://proton.semanticweb.org/2005/04/protonu#LandRegion
LaunchFacility	http://proton.semanticweb.org/2005/04/protonu#LaunchFacility
Leader	http://proton.semanticweb.org/2005/04/protonu#Leader
Legislation	http://proton.semanticweb.org/2005/04/protonu#Legislation
LocalCapital	http://proton.semanticweb.org/2005/04/protonu#LocalCapital
Magazine	http://proton.semanticweb.org/2005/04/protonu#Magazine
Man	http://proton.semanticweb.org/2005/04/protonu#Man
Manager	http://proton.semanticweb.org/2005/04/protonu#Manager
Market	http://proton.semanticweb.org/2005/04/protonu#Market
MediaBrand	http://proton.semanticweb.org/2005/04/protonu#MediaBrand
MediaCompany	http://proton.semanticweb.org/2005/04/protonu#MediaCompany
MediaProduct	http://proton.semanticweb.org/2005/04/protonu#MediaProduct
MemberOfParliament	http://proton.semanticweb.org/2005/04/protonu#MemberOfParliament
MilitaryAreas	http://proton.semanticweb.org/2005/04/protonu#MilitaryAreas
MilitaryConflict	http://proton.semanticweb.org/2005/04/protonu#MilitaryConflict
Minister	http://proton.semanticweb.org/2005/04/protonu#Minister
Ministry	http://proton.semanticweb.org/2005/04/protonu#Ministry
Money	http://proton.semanticweb.org/2005/04/protonu#Money
Month	http://proton.semanticweb.org/2005/04/protonu#Month
Mountain	http://proton.semanticweb.org/2005/04/protonu#Mountain
MountainRange	http://proton.semanticweb.org/2005/04/protonu#MountainRange
Movie	http://proton.semanticweb.org/2005/04/protonu#Movie
NaturalPhenomenon	http://proton.semanticweb.org/2005/04/protonu#NaturalPhenomenon
NewsAgency	http://proton.semanticweb.org/2005/04/protonu#NewsAgency
Newspaper	http://proton.semanticweb.org/2005/04/protonu#Newspaper

Ocean	http://proton.semanticweb.org/2005/04/protonu#Ocean
ofCountry	http://proton.semanticweb.org/2005/04/protonu#ofCountry
OfficialPosition	http://proton.semanticweb.org/2005/04/protonu#OfficialPosition
Park	http://proton.semanticweb.org/2005/04/protonu#Park
Parliament	http://proton.semanticweb.org/2005/04/protonu#Parliament
Peninsula	http://proton.semanticweb.org/2005/04/protonu#Peninsula
Percent	http://proton.semanticweb.org/2005/04/protonu#Percent
PeriodicalPublication	http://proton.semanticweb.org/2005/04/protonu#PeriodicalPublication
PhoneNumber	http://proton.semanticweb.org/2005/04/protonu#PhoneNumber
PieceOfArt	http://proton.semanticweb.org/2005/04/protonu#PieceOfArt
Plain	http://proton.semanticweb.org/2005/04/protonu#Plain
Planet	http://proton.semanticweb.org/2005/04/protonu#Planet
PoliticalEntity	http://proton.semanticweb.org/2005/04/protonu#PoliticalEntity
PoliticalParty	http://proton.semanticweb.org/2005/04/protonu#PoliticalParty
PoliticalRegion	http://proton.semanticweb.org/2005/04/protonu#PoliticalRegion
PopulatedPlace	http://proton.semanticweb.org/2005/04/protonu#PopulatedPlace
Premier	http://proton.semanticweb.org/2005/04/protonu#Premier
President	http://proton.semanticweb.org/2005/04/protonu#President
Profession	http://proton.semanticweb.org/2005/04/protonu#Profession
Province	http://proton.semanticweb.org/2005/04/protonu#Province
PublicCompany	http://proton.semanticweb.org/2005/04/protonu#PublicCompany
PublishedMaterial	http://proton.semanticweb.org/2005/04/protonu#PublishedMaterial
PublishingCompany	http://proton.semanticweb.org/2005/04/protonu#PublishingCompany
RadioStation	http://proton.semanticweb.org/2005/04/protonu#RadioStation
ReferenceLocation	http://proton.semanticweb.org/2005/04/protonu#ReferenceLocation
ReligiousLocation	http://proton.semanticweb.org/2005/04/protonu#ReligiousLocation
ReligiousOrganization	http://proton.semanticweb.org/2005/04/protonu#ReligiousOrganization
Report	http://proton.semanticweb.org/2005/04/protonu#Report
ResearchOrganization	http://proton.semanticweb.org/2005/04/protonu#ResearchOrganization
River	http://proton.semanticweb.org/2005/04/protonu#River
School	http://proton.semanticweb.org/2005/04/protonu#School
Sea	http://proton.semanticweb.org/2005/04/protonu#Sea
Season	http://proton.semanticweb.org/2005/04/protonu#Season
Ship	http://proton.semanticweb.org/2005/04/protonu#Ship
SoccerClub	http://proton.semanticweb.org/2005/04/protonu#SoccerClub
SocialAbstraction	http://proton.semanticweb.org/2005/04/protonu#SocialAbstraction
Spacecraft	http://proton.semanticweb.org/2005/04/protonu#Spacecraft
Sport	http://proton.semanticweb.org/2005/04/protonu#Sport
SportClub	http://proton.semanticweb.org/2005/04/protonu#SportClub
SportGame	http://proton.semanticweb.org/2005/04/protonu#SportGame
Stadium	http://proton.semanticweb.org/2005/04/protonu#Stadium
Star	http://proton.semanticweb.org/2005/04/protonu#Star
StockExchange	http://proton.semanticweb.org/2005/04/protonu#StockExchange
Street	http://proton.semanticweb.org/2005/04/protonu#Street
Team	http://proton.semanticweb.org/2005/04/protonu#Team
Telecom	http://proton.semanticweb.org/2005/04/protonu#Telecom
TimeZone	http://proton.semanticweb.org/2005/04/protonu#TimeZone
TransportFacility	http://proton.semanticweb.org/2005/04/protonu#TransportFacility
TVChannel	http://proton.semanticweb.org/2005/04/protonu#TVChannel
TVCompany	http://proton.semanticweb.org/2005/04/protonu#TVCompany
University	http://proton.semanticweb.org/2005/04/protonu#University
Valley	http://proton.semanticweb.org/2005/04/protonu#Valley
Vehicle	http://proton.semanticweb.org/2005/04/protonu#Vehicle
WeaponModelOrSystem	http://proton.semanticweb.org/2005/04/protonu#WeaponModelOrSystem
WebPage	http://proton.semanticweb.org/2005/04/protonu#WebPage
Woman	http://proton.semanticweb.org/2005/04/protonu#Woman
hasWeight	http://proton.semanticweb.org/2005/04/protonkm#hasWeight
Abbreviation	http://www.ontotext.com/kim/2005/04/kimlo#Abbreviation
CountryAdj	http://www.ontotext.com/kim/2005/04/kimlo#CountryAdj
MilitaryTitle	http://www.ontotext.com/kim/2005/04/kimlo#MilitaryTitle
PersonFirstFemale	http://www.ontotext.com/kim/2005/04/kimlo#PersonFirstFemale
PoliceTitle	http://www.ontotext.com/kim/2005/04/kimlo#PoliceTitle
TimeModifier	http://www.ontotext.com/kim/2005/04/kimlo#TimeModifier
Title	http://www.ontotext.com/kim/2005/04/kimlo#Title

Table 9: Proton Classes