EU-IST Integrated Project (IP) IST-2003-506826 SEKT

SEKT: Semantically Enabled Knowledge Technologies

D2.1.1 Ontology-Based Information Extraction (OBIE) v.1

Yaoyong Li (University of Sheffield)
Kalina Bontcheva (University of Sheffield)
Mike Dowman (University of Sheffield)
Ian Roberts (University of Sheffield)
Hamish Cunningham (University of Sheffield)

Abstract.
EU-IST Integrated Project (IP) IST-2003-506826 SEKT
Deliverable D2.1.1 (WP2)

This deliverable presents an SVM-based algorithm for IE and experiments on several benchmark datasets. The results showed that our system is comparable to other state-of-the-art systems on both traditional IE and adaptive IE tasks. We investigated two feature weighting schemes, the impact of different NLP features on the performance of the learning algorithm, and the importance of the SVM parameters. Results are reported both on traditional IE tasks such as named entity recognition and slot filling, and on adaptive IE and hierarchical named entity learning. Directions for future work are discussed in the conclusion.

Keyword list: Ontology-based Information Extraction, Machine Learning, Adaptive IE

Document Id. SEKT/2004/D2.1.1/v1.0
Project SEKT EU-IST-2003-506826
Date January 25, 2004
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
Techikerstraße 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

Information Extraction (IE) systems fall in two broad categories: manually engineered ones (frequently rule-based) (e.g., [MTU$^+$01]) and machine learning ones (e.g. [BSW99]). Manually engineered systems have good performance, but are hard to adapt by non-specialists, e.g., end users. Consequently, engineered systems are expensive to develop, enhance, and port to new domains, as changes in the user requirements can only be implemented by specialists. On the other hand, machine learning systems tend to achieve comparable performance but, once created, their behaviour can be changed by non-specialists by them providing further training data. Therefore machine learning systems are preferred in applications where the portability and changing requirements occur. In SEKT we adopt the machine learning approach because ontologies change over time and more portable systems are required. This is particularly true if the ontologies contain fewer concepts but with thousands of instances for each. Engineering manually a high performance system in this case might be more difficult than having a learning system being trained in the background while the user is annotating texts. Once the performance has reached a satisfactory level (as defined by the user), the learning algorithm can start suggesting annotations and the user then only needs to correct them. Ultimately, the user can decide to run the system in a fully automatic mode. This is often referred to as *mixed-initiative IE* or *adaptive IE* (e.g., [DAH$^+$97]). The goal of the work in SEKT is to create such tools for ontology-based IE. In the first year we focused on developing a state-of-the-art learning system, then it will be improved further over the course of the project. The mixed-initiative environment will also be implemented and extensive experiments will be carried out. We plan a comparison with a manually engineered ontology-based IE system, i.e., KIM [PKK$^+$04].

This deliverable presents an SVM-based algorithm for IE and experiments on several benchmark datasets. The results showed that our system is comparable to other state-of-the-art systems on both traditional IE and adaptive IE tasks. In comparison to other similar SVM-based algorithms, our algorithm is simpler, i.e., it needs a smaller number of SVM classifiers per category than the other two systems discussed respectively in [IK02] and [MMP03]. GATE-SVM also obtained better results than the SVM-based system in [MMP03] on the CoNLL-2003 corpus.

Our algorithm uses an uneven margin parameter, which we showed to be particularly useful for adaptive information extraction on a small amount of training data.

We also investigated two weighting schemes for the features of the surrounding words and showed that the reciprocal weighting scheme performed better than the commonly used equally weighting. We also investigated three post-processing procedures: from using the SVM outputs for begin and end tags separately to selecting the highest probability label based on the output of all SVM classifiers. We found that the probability scheme gave best results.

We also carried out experiments using various combinations of features in order to systematically investigate the effects of different NLP features on the performance of the learning algorithm. The NLP features produced by the general purpose modules from GATE were used and none of them are domain specific, unlike some used by other systems. Words provided most information and basic linguistic features such as capitalisation and lemmas were often quite helpful, however part-

of-speech information was not useful. Both the general ANNIE gazetteers and rule-based named entity recogniser were helpful. Despite the fact that GATE provides document formatting information, e.g., HTML tags, such text-specific features were not utilised in our current experiments.

The main reason for applying our machine learning algorithms to corpora like CMU seminars and jobs, instead of an ontology annotated corpus, comes from the lack of corpora for IE, annotated with rich ontological information. Other related projects, e.g., DOT.KOM[1], have also used the seminars and jobs corpora instead.

This work goes one step further by using the ACE corpus which provides hierarchically structured named entities and is thus closest in spirit to the data needed for ontology-based IE. Therefore, our first experiments on using statistical methods for ontology-based IE used this corpus.

A still ongoing recent effort is the Pascal Challenge on evaluating machine learning for information extraction[2]. As part of this initiative, an annotated corpus of workshop calls for papers (CFPs) has been produced. However, similar to the jobs postings, this corpus uses a very flat ontology of approximately 16 classes.

Future work on ontology-based IE in SEKT will fill this gap by using the newly created ontologically annotated corpus (see D2.5.1 Evaluation Tools and Corpora). Another strand of future work on OBIE will be concerned with the creation of user-friendly tools for training the system., possibly augmented with information from the Internet, e.g., the Pankow system [CHS04].

[1] http://nlp.shef.ac.uk/dot.kom/resources.html

[2] http://nlp.shef.ac.uk/pascal/

Contents

1 Introduction

Information Extraction (IE) is the process of automatic extraction of information about pre-specified types of events, entities or relationships from text such as newswire articles or Web pages (see [Cun05] for a comprehensive overview of IE). A lot of work have been done on named entity recognition, a basic task of IE, which aims to classify the proper nouns and/or numerical information in documents. Actually most IE tasks can be viewed as the task of recognising some information entities from the text. IE can be useful in many applications, such as information gathering in a variety of domains, automatic annotations of web pages for semantic web, and knowledge management.

Machine learning techniques have been used for IE and achieved state of the art results. In the applications of machine learning to IE, a learning algorithm usually extracts a model from a set of documents which have been manually annotated by the user. Then the model can be used to extract information from new documents. Usually the algorithm would learn a more accurate model if given more training examples. However, manual annotation is a time-consuming process. Hence, in many applications the so called adaptive or mixed-initiative learning is desirable (see e.g., Alembic [DAH$^+$97], Amilcare [CW03]). In a mixed-initiative IE system, a few documents are manually annotated first (i.e., the user has the initiative to begin with). The system learns an initial model from this small pool of annotated examples. Then the model is applied to tag new documents (the system starts having some initiative by suggesting the tags) and the results are corrected by the user. Then the system updates the model based on the user's corrections, and the process continues until the user is satisfied with the system performance and allows it to work fully automatically. In order to lower the overhead of training a learning system, this kind of human-machine interactive approach is crucial for building an efficient and flexible IE system. Therefore, an important part of this work is focused on evaluating our learning algorithm on growing amounts of data, starting from a small set of annotated documents.

Machine learning algorithms for IE can be classified broadly into two main categories: rule learning and statistical learning. The former induces a set of rules from a training set, while the later learns a statistical model or classifiers. Support Vector Machines (SVM) is a general supervised machine learning algorithm, that has achieved state of the art performance on many classification tasks, including NE recognition (see e.g. [IK02], [MMP03]). [IK02] compared three commonly used methods for named entity recognition – the SVM with quadratic kernel, maximal entropy method, and a rule based learning system, and showed that the SVM based system performed better than the other two. In our view, the comparison between different learning methods in [IK02] is more informative than the comparison in, e.g., the CoNLL-2003 share task (see [SM03]), because the former used both the same corpus and the same features for all the systems, while in the later different systems used the same corpus but different features.[3] [IK02] also described an efficient implementation of the SVM with quadratic kernel. [MMP03] used a lattice-based approach to named entity recognition and employed the SVM with cubic kernel to compute transition probabilities in a lattice. Their results on CoNLL2003 shared task were comparable to other systems but were not the best ones.

[3] The still ongoing Pascal Challenge in evaluation of machine learning methods for IE aims to provide a corpus and a pre-defined set of features, so different algorithms can be compared better (http://nlp.shef.ac.uk/pascal/).

This report describes an SVM-based learning algorithm for IE and present detailed experimental results. In contrast to previous similar work, our SVM model (see Section 2) uses an uneven margins parameter which has been shown [LST03] to improve the performance for document categorisation (especially for small categories). Detailed experiments to investigate different experimental settings of the SVM based algorithm on several benchmark datasets was carried out (see Sections 3 and 4). We also investigate the effect of different NLP features which were produced by the Natural Language Processing (NLP) system ANNIE, which is part of the open-source GATE infrastructure [CMBT02]. The experimental datasets were chosen to enable thorough comparisons between our approach and other state of the art learning algorithms (see Section 4.2). The algorithm was also evaluated in simulated mixed-initiative settings, where only a small number of documents was given for learning initially and then more and more documents were provided incrementally. Section 6 covers related work.

2 The SVM based Learning Algorithm

We used a variant of the SVM, the SVM with uneven margins [LST03], which has a better generalisation performance than the original SVM on datasets where the positive examples are much less than the negative ones. The uneven margins parameter has been shown previously to facilitate document classification on unbalanced training data (see [LST03]). Given that IE classification tasks, particularly when learning from small data sets, often involve unbalanced data, we decided to use SVM with uneven margins, instead of the original SVM algorithm.

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 following quadratic optimisation problem:

$$\begin{array}{rl} \text{minimise}_{\mathbf{w},\, b,\, \xi} & \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{m} \xi_i \\ \text{subject to} & \langle \mathbf{w}, \mathbf{x}_i \rangle + \xi_i + b \geq 1 \quad \text{if } \; y_i = +1 \\ & \langle \mathbf{w}, \mathbf{x}_i \rangle - \xi_i + b \leq -\tau \quad \text{if } \; y_i = -1 \\ & \xi_i \geq 0 \qquad \text{for } \; i = 1, ..., m \end{array} \tag{1}$$

In these equations, τ is the uneven marginal parameter which is the ratio of negative margin to the positive margin in the classifier and is equal to 1 in the original SVM.

2.1 Use of Context in the Feature Vectors

When statistical learning methods are applied to IE tasks, they are typically formulated as classification, i.e., each word in the document is classified as belonging or not to one of the target classes (e.g., named entity tags). The same strategy was adopted in this work, which effectively means that each word is regarded as a separate instance by the SVM classifier. First a input vector is formed, based on a large number of features. Since in IE the context of the word is usually as important as the word itself, the features in the input vector come not only from the given word to be classified, but also from preceding and following words. In our experiments the same number of left and right words was taken as a context. In other words, the current word was at the centre of a window of words 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 words: the three preceding, the current, and the 3 following words.

Due to the use of a context window, the SVM input vector is the combination of the feature vector of the current word and those of the neighbouring words. The feature vector derived from a word is a long sparse vector. First, the algorithm collects all possible features from the training documents. Each feature (e.g. a token or a part-of-speech (POS) category) corresponds to one dimension in the feature vector. So the vectors tend to have thousands of dimensions, with only a small number of components being nonzero. We set the value of nonzero components to 1.

As the input vector of the SVM comes from the words in a window surrounding the current word, we can weight those feature vectors from different words according to our knowledge about the relative importance of the neighbouring words. Two weighting schemes for the feature vectors from neighbouring words were investigated. The first is *equal weighting*, which keeps every nonzero component of the feature vector as 1 in the combined input vector, i.e., treats all neighbouring words as equally important. The second weighting scheme is the *reciprocal scheme*, which weights the surrounding words reciprocally to the distance to the word in the centre of the current window, reflecting the intuition that the nearer a neighbouring word is, the more important it is for classifying the given word. Formally it means that the nonzero components of the feature vector corresponding to the jth right or left neighbouring word are set to be equal to $1/j$ in the combined input vector. Therefore, we also refer to this scheme as $1/j$ weighting.

2.2 Post-processing

In our system we train two SVM classifiers for each type of entity – one classifier for the start and another one for the end word. One word entities are regarded as both start and end. In contrast, [IK02] trained four SVM classifiers for each named entity type – besides the two SVMs for start and end (like ours), also one for middle words, and one for single word entities. They also trained an extra SVM classifier to recognise words which do not belong to any named entity. [MMP03] trained an SVM classifier for every possible transition of tags so, depending on the number of entities, that may lead to a large number of SVM classifiers.

As our SVM classifiers only identity the start or end word for every target class, some post-processing is needed to combine these into a single tag. We implemented a module with *three different stages* to post-precess the results from SVM classifiers:

- The *first stage* uses a simple procedure to guarantee the consistency of the recognition results. It scanned a document to remove start tags without matching end tags and end tags without preceding start tags.

- The *second stage* filters out candidate entities from the output of the first stage, based on their length. Namely, the tags of a candidate entity are removed if the entity's length (the number of words) is not equal to the length of any entity of the same type in training set (a similar method was used in [FK00]).

- In contrast with the above two stages where each candidate entity is considered separately, the *third stage* puts together all possible tags for a given word and choses the best one.

In detail, the output x of the SVM classifier (before thresholding) was first transferred into a probability via the Sigmoid function $s(x) = -0.5 + 1/(1 + exp(-\beta x))$ where β was set as 2.0 in our experiments (also see [IK02] and [MMP03]). Then a probability for an entity candidate was computed via $2 * s(x_s) * s(x_e)$, where x_s and x_e are the outputs of the SVM classifier for the start and end words of the candidate, respectively. Finally, for each given word, the probabilities for all possible tags were compared to each other and the tag with the highest probability P_h is assigned if P_h is greater than 0. Otherwise no tag is assigned to the word.

In our implementation the user can choose which of these three stages they want to be used during training, i.e., only the first, the first and the second, and all three. Note that both [IK02] and [MMP03] used a Viterbi search algorithm as a post-procedure for their SVM classifiers, which corresponds to our third stage.

3 The experimental datasets

The system was evaluated on three corpora covering different IE tasks – named entity recognition (CoNLL-2003) and template filling or scenario templates [SAI98] (seminars and jobs corpora). There were several reasons for choosing these corpora. Firstly, CoNLL-2003 provides the most recent evaluation results of many machine learning algorithms on named entity recognition. Secondly, the seminars and jobs corpora have also been used recently by many learning systems, both wrapper induction and more linguistically oriented ones (see Section 6 for a detailed discussion). Thirdly, the CONLL-2003 corpus differs from the other two corpora in two important aspects: (i) in CONLL-2003 there are many entities per document, whereas the jobs and seminar corpora have only a small number per document; (ii) CONLL-2003 documents are mostly free text, whereas the other two corpora contain semi-structured documents. Therefore, the performance of our SVM algorithm was evaluated thoroughly on these three corpora as our goal was to design a versatile approach, with state-of-the-art performance both on domain-independent IE tasks (e.g., named entity recognition) and domains-specific ones (e.g., template filling).

In more detail, the first corpus is the English part of the CoNLL-2003 shared task dataset — language-independent named entity recognition[4]. This corpus consists of 946 documents for training, 216 documents for development (e.g., tuning the parameters in learning algorithm), and 231 documents for evaluation (i.e., testing), all of which are news articles taken from the Reuters English corpus (RCV1) [LYRL04]. The corpus contains four types of named entitis — person, location, organisation and miscellaneous names.

The other two corpora are the CMU seminar announcements and the software job postings[5], in

[4] See http://cnts.uia.ac.be/conll2003/ner/

[5] Available from http://www.isi.edu/info-agents/RISE/repository.html.

both of which domain-specific information was extracted into a number of slots. The seminar corpus contains 485 seminar announcements and four slots – starting time (stime), end time (etime), speaker and location of a seminar. The job corpus includes 300 computer related job advertisements and 17 slots such as title, salary and recruiter of the job and computer language, and application and platform required by job.

Table 1 shows the statistics for the CoNLL-2003, seminars and jobs datasets, respectively. We can see that the non-annotated words are much more than the annotated words, particularly for domain-specific datasets like seminar announcements and software job postings.

Table 1: Number of examples for each entity/slot type, together with the number of non-tagged words, in CoNLL-2003 corpus, seminars announcements, and software jobs postings, respectively.

Conll03	LOC	MISC	ORG	PER	Non-entity	
Training set	7140	3438	6321	6600	191627	
Test-a set	1837	922	1341	1842	47926	
Test-b set	1668	702	1661	1617	43654	
Seminars	Stime	Etime	Speaker	Location	Non-entity	
	980	433	754	643	157647	
Jobs	Id	Title	Company	Salary	Recruiter	State
	304	457	298	141	312	462
	City	Country	Language	Platform	Application	Area
	659	345	851	709	590	1005
	Req-years-e	Des-years-e	Req-degree	Des-degree	Post date	Non-entity
	166	43	83	21	302	127302

Machine learning systems typically separate the corpus into training and test sets. Since the CoNLL-2003 corpus already has the training, development and test set pre-specified, the system is trained on the training set, different experimental settings are tested on the development set, and the optimal ones are used to obtain the final results on the test set, which are then used for comparison with other systems.

The other two corpora do not provide such different sets, therefore training and testing need to be carried out differently. In our experiments we opted for splitting the corpora into two equal training and test sets by randomly assigning documents to one or the other[6]. In order to obtain more representative results, we carried out several runs and the final results were obtained by averaging the results from each run. Many of the learning systems evaluated on these corpora used the same approach and we adopted it to facilitate comparison (see Section 4.2).

All corpora were also pre-processed with the open-source ANNIE system, which is part of GATE [CMBT02]. This enabled us to supply our system with 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,

[6] As the total number of documents in the seminar corpus is 485, we randomly split the dataset into 243 training documents and 242 testing ones.

punctuation), lemma, part-of-speech (POS) tag, gazetteer class, and named entity type according to ANNIE's rule-based recogniser. The following section discusses the systematic experiments that investigated the effects of each of these NLP features.

4 Experimental results

This section presents the experimental results on the three datasets described above. As already discussed in Section 2, two SVM classifiers were trained for each entity and slot filler, one for the start and one for the end words. The resulting models were then run on the test set and the post-processing procedures described in Section 2 were applied. All experiments used the SVM package SVMlight version 3.5[7]. Unless stated otherwise, the default values of the parameters in SVMlight 3.5 were used.

The results below are reported using the F_1-measure, which is. the harmonic mean of precision and recall. In other words, $F_1 = (2 * precision * recall)/(precision + recall)$, where $precision$ is the percentage of correct entities found by the system and $recall$ is the percentage of entities in the test set which are found by the system. A tag is considered correct if it matches exactly the human-annotated tag, both in terms of its type and its start and end offset in the document.[8]

The overall performance of the algorithm on a given corpus can be obtained in two different ways. One is the so called *macro-averaged* F_1, which is the mean of F_1 of all the entity types or slots in the corpus. The other is the *micro-averaged* measure[9], obtained by adding together the recognition results on all entity types first and then computing precision, recall, and f-measure. Some researchers argue that the macro-averaged measure is better than the micro-averaged one (see e.g. [YL99]), because the micro-averaged measure can be dominated by the larger classes so that it reflects less the performance of the algorithm on smaller classes. On the other hand, if all classes are of a comparable size, as is often the case in IE datasets, then the macro-averaged measure is not very different from the micro-averaged one. In addition, commonly used IE evaluation tools such as the MUC scorer [SAI98] tend to use the micro-averaged measure. The tables below report micro-averaged F-measure on the CONLL corpus, because it was used for reporting the overall system results. We also use macro-averaged F-measure when we need to obtain an overall measure of the system's performance, e.g., for the purpose of establishing the impact of different parameters on the system's performance. The majority of systems evaluated on the jobs and seminars corpora only reported per slot F-measures, without overall results, so we will adhere to this convention, when comparing our scores to other systems.

[7] Available from http://www.joachims.org/svm_light

[8] Although the results taking account both exact and partial match are informative in some applications, we only report results using exact match in order to make our results comparable to those reported on the CoNLL-2003 corpus, as well as in other previous work. In order to get an estimate of the influence of partial matching, we carried out some experiments which showed that partial match resulted in additional 0.01 – 0.03 F_1 over the results from exact match only.

[9] See http://www.itl.nist.gov/iaui/894.02/related_projects/muc/muc_sw/muc_sw_manual.html

4.1 Influence of Different SVM Parameters and NLP Features

As part of the system evaluation, we conducted a series of experiments to investigate the influence of different SVM parameter settings as well as different combinations of NLP features. The first group of experiments looked into different window sizes. Then we tested different SVM kernels and values of the uneven margins parameter. Next different combinations of NLP features were applied and finally we compare the two weighting schemes introduced in Section 2.

In order to avoid testing each possible setting against all others, the optimal setting obtained from one group of experiments was used in subsequent experiments. However, while keeping the number of experiments down, this kind of sequential optimisation may not result in the most optimal parameter settings. For example, the optimal window size from the first group of experiments using linear kernel may not be optimal for the later experiments using quadratic kernel. Hence, the optimal setting obtained at each stage may not be the global optimal value, although we believe they are near each other.

Window size. We first did experiments using different window sizes on the three datasets. All NLP features, discussed in Section 3 were used together with word and capitalisation information. Table 2 presents the results for window size between 0 to 6. As can be seen, the results improve substantially when the window size changes from 0 to 1, which confirms that context is important in IE. The results also show that different datasets have different optimal window sizes – 5 for the seminars, 3 for the jobs, and 2 for the CoNLL-2003 dataset. Therefore, all subsequent experiments used the window size most optimal for the given dataset. In actual fact, even within the same dataset, different entities/slots seem to have different optimal window sizes. For example, although 5 is the best window size for the overall performance on the seminars corpus, the f-measure for the location slot at window size 3 (0.845) is higher than the one for window size 5 (0.816). However, in order to simply the experimental settings, the overall optimal window sizes were used on each corpus.

Table 2: Different window sizes: macro-averaged F_1 on the three datasets. SVM with linear kernel, equal weighting, and all NLP features were used.

Dataset	0	1	2	3	4	5	6
CoNLL03 shared task	0.732	0.863	0.873	0.863	0.859	0.853	0.843
Seminar announcements	0.261	0.645	0.735	0.809	0.840	0.842	0.824
Job advertisements	0.479	0.748	0.754	0.774	0.748	0.762	0.743

SVM Model selection. The next set of experiments focused on testing different parameters of the learning algorithm. We used the same features as above, equal weighting, and the optimal window size for each corpus, that was determined in the previous experiments. Three types of SVM kernels were compared, namely, linear, quadratic, and cubic kernels (see e.g. [IK02] and [MMP03]). The results are presented in Table 3, which shows once again that different datasets have different optimal kernels. Among the three kernels, the quadratic kernel is the best for the CoNLL-2003 dataset, while the linear kernel is the best for the other two datasets.

An SVM parameter that also affects the performance is the uneven margins parameter τ (see Sec-

Table 3: Results of three kernels for the SVM: macro-averaged F_1 on the three datasets.

Dataset	Linear Kernel	Quadratic kernel	Cubic kernel
CoNLL03 shared task	0.873	0.883	0.875
Seminar announcements	0.842	0.827	0.795
Job advertisements	0.774	0.737	0.699

tion 2). As already discussed, τ is the ratio of negative margin to positive margin. The optimal value of τ is dependent on the dataset. We did experiments with two values for τ: $\tau = 0.4$ and $\tau = 1$ (as in the original SVM), using the linear and quadratic kernels, as they were the ones that performed best. The results are presented in Table 4. We can see that $\tau = 0.4$ gave better results than the original SVM with $\tau = 1$ in 3 out of 4 cases on the seminars and jobs corpora but had slightly worse result on the CoNLL-2003 dataset. Nevertheless, for uniformity it will be used in the rest of the experiments, as it was the better setting on two out of the three corpora. Table 4 also shows that the quadratic kernel yields better results than the linear kernel for $\tau = 0.4$ on the CoNLL-2003 and seminars datasets, whereas the linear kernel is better for the jobs corpus.

Table 4: The SVM uneven margins parameter τ. Macro-averaged F_1 results with linear and quadratic kernels on the three datasets.

Dataset	Linear kernel		Quadratic kernel	
	$\tau = 1$	$\tau = 0.4$	$\tau = 1$	$\tau = 0.4$
CoNLL03	0.873	0.870	0.883	0.877
Seminar	0.842	0.835	0.827	0.854
Job	0.774	0.789	0.737	0.781

NLP features. Table 5 presents the experimental results using different NLP features for three datasets. First, only the word itself is used, then the NLP features were added one by one, from the simpler to the more complicated. The NLP features include case information, simple categorisation information of the token, called Tokenkind in the tables below, (i.e., number, punctuation mark, word), lemma, part-of-speech (POS) tag, semantic category obtained from the ANNIE gazetteer, and general entity type from the named entity recognition system ANNIE. Note that none of these features is domain or corpus specific, e.g., whether the word appears in a specific place in the seminar header. Instead the NLP features were produced by the general-purpose NLP components in GATE.

We found that words provided most information and that capitalisation and lemma were also helpful. The token type information (i.e., number, punctuation, etc.) was quite helpful in two of the three datasets. POS information is helpful for seminar announcements but has negative impact on performance on the other two datasets. Both the general ANNIE gazetteer and the outputs of the named entity recogniser were helpful, although they did not help increase the performance as much as expected.

Using different kinds of gazetteers as features. The GATE gazetteers provide some seman-

Table 5: Different NLP features: macro-averaged F_1 on CoNLL-2003, seminars and jobs corpora, respectively. The quadratic kernel was used and the uneven margin parameter $\tau = 0.4$. WT refers to the two features Word and Tokenkind, and WCTL refers to four features, namely Word, Case information, Tokenkind and Lemma.

Features	Word	Word+Case	Word+Case	WCTL	WCTL	WCTL	WCTL
			+Tokenkind		+Pos	+Gaz	+Gaz+Entity
CoNLL03	0.799	0.853	0.858	0.864	0.857	0.871	0.883
Seminars	0.781	0.824	0.839	0.845	0.848	0.860	0.861
Jobs	0.782	0.778	0.785	0.789	0.777	0.780	0.796

tic information, which was showed above to be helpful. The CoNLL-2003 share task provided a gazetteer for the CoNLL-2003 corpus. We experimented whether using both the GATE's and CoNLL-2003's gazetteers was beneficial. Table 6 presents the results for different combinations of the two gazetteers. As in the previous experiment, we trained the SVM classifiers using the training set and tested them on the development set (test-a). Surprisingly, the performance deteriorated whenever the CoNLL-2003 gazetteer was used, in particular for two types of named entities, Person and Organisation. On the other hand, generally speaking, a gazetteer has been found helpful for better generalisation during learning for information extraction. Therefore some explanation was needed to account for these unexpected results.

Table 6: Different NLP features from gazetteers: macro-averaged F_1 on the CoNLL-2003 dataset. The quadratic kernel was used and the uneven margin parameter $\tau = 0.4$. Gaz1 is the gazetteers provided by CoNLL-2003 corpus and Gaz2 is the gazetteer from GATE system. Entity means the named entity type from the ANNIE. Finally WCTL denotes the four features: Word, Case information, Tokenkind, and Lemma.

	WCTL	WCTL	WCTL	WCTL+GATE gaz
		+Conll gaz	+GATE gaz	+Conll gaz
Person	0.8892	0.5772	0.9050	0.5835
Location	0.8995	0.8881	0.8985	0.8894
Organisation	0.8228	0.7478	0.8415	0.7481
Miscellaneous Names	0.8429	0.8403	0.8427	0.8389
Average	0.8636	0.7633	0.8712	0.7650

To uncover the reason why the CoNLL-2003 gazetteer was harmful, we computed the correlation between the named entity tags derived from either the CoNLL-2003 gazetteer, or the GATE gazetteer, and the labels in the CoNLL-2003 corpus. Table 7 lists the correlations of the tag Person for the training, test-a and test-b sets of the CoNLL-2003 corpus, respectively. As can be seen, for tag Person the entries in the CONLL gazetteer are closely correlated with the label tags in the training set – more than 99% CONLL gazetteer entries are label tags and vice versa.

As a result, a good recogniser learned from the training set would exploit the nearly perfect correspondence between CONLL gazetteer entries and label tags and regard the gazetteer entries as a key feature to identify the named entity Person. Such a strategy would achieve more than 99% performance on the training set. In fact, as we will see below from the results on the test-a and test-b datasets, it seemed that our SVM-based recogniser did use this kind of strategy. However, unfortunately, the perfect correspondences between CONLL gazetteer entries and Person tags does not exist in either the test-a set or test-b set. For example, only about 47.2% of the label tags are also gazetteer entries, although 97.2% of CONLL gazetteer entries are label tags. The recall of the named entity Person is 43.9% and 16.0% respectively on test-a and test-b, which are similar to the numbers of conll/label in Table 7.

To summarise, the bad result of the Person recogniser on test-a and test-b datasets may be caused by the fact that the recogniser utilised the perfect correspondence between CONLL gazetteer entries and label tags for Person in the training set and the correspondence deteriorated on the test-a and test-b sets.

On the other hand, the correlation between the GATE gazetteer entries and label tags remains similar for the training, test-a and test-b sets and using GATE gazetteer was helpful as showed in Table 6.

In order to verify our observation further, we did another experiments by using the test-a for learning a Person recogniser and testing it on the test-b set, as the correlations between CONLL gazetteer entries and label tags are similar for the two sets. The results are shown in Table 8 and they demonstrate that the CoNLL-2003 gazetteer was helpful. For the named entity Organisation we have the similar observation to the case of the entity Person.

Table 7: The correlations between the named entity tags derived from three sources – the GATE gazetteer, the CoNLL-2003 gazetteer, and the labels of CoNLL-2003 corpus, respectively. Only the numbers for named entity Person are shown for training, test-a and test-b sets of CoNLL-2003 corpus, respectively.

	label/CONLL Gaz	CONLL Gaz/label	label/GATE Gaz	GATE Gaz/label
training	99.1%	99.2%	86.9%	34.2%
test-a	97.2%	47.2%	88.8%	33.6%
test-b	93.0%	18.9%	88.7%	33.1%

Table 8: Experiments for different combinations of the CoNLL-2003 and GATE gazetteers: test-a for training test-b for test.

Features	WCTL +CONLL gaz	WCTL +GATE gaz	WCTL +CONLL gaz	WCTL+GATE gaz
Averaged F_1	0.7158	0.7811	0.7398	0.7910

Two weighting schemes. Table 9 presents the results for two weighting schemes – the *equal weighting* and the *reciprocal weighting* of neighbouring words. In the former, all the features of the current word as well as neighbouring words are weighted equally. In the latter, the features of neighbouring word are weighted reciprocally to the distance to the current word. Table 9 shows that the reciprocal scheme produced better results than equal weighting on the CoNLL-2003 data in two NLP feature sets, and performed better in one of the two feature sets on the other two corpora. Consequently, $1/j$ weighting scheme is used for the subsequent experiments on the three corpora.

Table 9: Two weighting schemes: macro-averaged F_1 on the three datasets. The results are on two sets of NLP features, respectively. The first feature set denoted by WCTL includes the four NLP features, word, case information, tokenkind and Lemma. The second feature set added the gazetteer and named entity recognition information to the first feature set.

Dataset	WCTL		WCTL + Gaz + NE	
	Equal weighting	$1/j$ weighting	Equal weighting	$1/j$ weighting
CoNLL03	0.863	0.873	0.883	0.890
Seminar	0.845	0.838	0.861	0.867
Job	0.789	0.804	0.796	0.787

Three post-processing procedures

Table 10 presents results for the three post-processing procedures discussed in Section 2. In brief, the first procedure removes the spurious start or end tags. The second procedure evaluates the lengths of candidate tags and removes a candidate if its length is not equal to the length of any tag of the same type in the training set. The third procedure considers all tags simultaneously and outputs the one with the highest probability. Table 10 presents the results of the three procedures with two NLP feature sets on the three corpora. The first feature set includes the basic NLP features: the word itself, case information, token kind and lemma. The second one includes two additional features based on the gazetteer and the named entity recognition system ANNIE.

The results in Table 10 show that the second procedure has better performance than the first on all datasets. However, the third procedure is better than the second in most cases. Therefore, subsequent experiments will use the third post-processing procedure, i.e., output the tag with the highest probability.

Table 10: Three post-processing procedures: macro-averaged F_1 on three datasets. The results are on two sets of NLP features, respectively. Proc1, Proc2 and Proc3 denote the three post-processing procedures described in the text.

Dataset	WCTL			WCTL+Gaz+Entity		
	Proc1	Proc2	Proc3	Proc1	Proc2	Proc3
CoNLL03	0.8731	0.8757	0.8756	0.8895	0.8912	0.8918
Seminar	0.8382	0.8400	0.8400	0.8675	0.8677	0.8686
Job	0.8045	0.8054	0.8050	0.7871	0.7889	0.7891

4.2 Comparison to other systems

In this subsection we compare our system with others on the three datasets. Since our system uses the NLP features produced by GATE and the learning algorithm based on SVM, we call our system **GATE-SVM**. In the experiments described in this subsection we would use the same settings as other systems in order to make a fair comparison.

Note that [SM03] presented the estimated significance boundaries for the results of many systems on the CoNLL-2003 corpus, which was computed via bootstrap sampling method. The significance boundaries can be used to determine if one result is significantly different from other results. We used the significance interval presented in [SM03] when we compare our results with others on the CoNLL-2003 dataset. However, unfortunately, the significance boundaries are not available for the previous results on the seminars and jobs corpora. Therefore, we estimated the significance boundaries for our results on the two datasets by also using the bootstrap sampling method (see [DH97]). In detail, for one experiment, 1999 random repetition samples of documents had been chosen and the distribution of F_1 in these samples was assumed to be the distribution of the performance of the experiment. As in [SM03], we chose the significance boundaries as the left and right boundaries of the interval with centered 90% of the distribution of the F_1 value.

4.2.1 Named Entity Recognition

We first evaluated our system on the CoNLL-2003 dataset. Since there was a development set for tuning the learning algorithm, we tried different settings to obtain the best performance on the development set. Once again we only tested the different SVM kernel types, the window sizes, and the uneven margin parameter τ. We found that the quadratic kernel, window size 4 and $\tau = 0.5$ produced best results on the development set. The $1/j$ weighting scheme and the probability post-processing procedure were used.

Table 11: Comparison with other systems on CoNLL-2003 shared task: F-measure on each entity type and the overall micro-averaged F-measure. The macro-averaged F-measure is included for comparison. Test-a denotes the development set and test-b – the test set.

System	test set	LOC	MISC	ORG	PER	MA F_1	Overall
GATE-SVM	test-a	**0.9370**	**0.8613**	**0.8700**	**0.9303**	**0.8996**	**0.9083**
	test-b	**0.8925**	**0.7779**	**0.8229**	**0.9092**	**0.8506**	**0.8630**
Best one	test-a	0.9612	0.8906	0.9024	0.9660	0.9301	0.9387
	test-b	0.9115	0.8044	0.8467	0.9385	0.8753	0.8876
Another	test-a	0.9375	0.8602	0.8590	0.9391	0.8990	0.9085
SVM System	test-b	0.8877	0.7419	0.7900	0.9067	0.8316	0.8467

Table 11 presents the best results of our algorithm on the CoNLL-2003 dataset, together with the results of the top system in the CoNLL-2003 share task evaluation [FIJZ03] and another participating SVM-based system [MMP03]. Our system outperformed the other SVM-based system but is slightly worse than the best result. Compared to the summarised results in [SM03], our overall result is slightly better than the third best system that participated in the original CoNLL-2003 evaluation but there is no significant difference between the two results.

Experiments on the News dataset

Table 12: 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. news	1081	2030	858	701	78	86
UK news	897	816	811	635	94	54

We also did experiment on the News dataset. The news dataset consists of three types of news documents – business news (93 docs), international political news (116 docs), and UK political news (114 docs). All documents are annotated with six kinds of named entities – Person, Location, Organisation, Date, Money and Percent (for statistics see Table 12. In these experiments we used the same experimental settings as on the CoNLL-2003 dataset, i.e., window size 4, a quadratical kernel, uneven margin parameter $\tau = 0.5$, the $1/j$ weighting, and the probability post-processing procedure. The goal was to establish whether GATE-SVM could achieve comparable performance on another similar corpus.

Table 13: Experiments on news dataset using different types of documents as training and test set, respectively: F_1 on individual type of entity and the overall performances.

	Person	Location	Organisation	Date	Money	Percent
Two types of news for training, another type for test						
UK news	0.927	0.938	0.805	0.922	0.989	0.991
Int. news	0.911	0.939	0.854	0.931	0.981	0.940
Business news	0.905	0.910	0.860	0.823	0.937	0.978
10-fold CV on every of three types and on the mixed, respectively						
Business news	0.948	0.912	0.904	0.897	0.974	0.981
Int. news	0.930	0.951	0.851	0.923	0.960	0.793
UK news	0.914	0.918	0.830	0.910	0.988	0.822
Mixed data	0.926	0.936	0.864	0.899	0.945	0.976
Different training set, the second half of News for testing						
CONLL’03	0.493	0.855	0.773	-	-	-
CONLL’03+ first half of News	0.875	0.930	0.841	-	-	-
Second half of News only	0.932	0.934	0.847	-	-	-

Table 13 presents the experimental results for the news corpus, using the following features: the word itself, case information, token kind, lemma, GATE gazetteer information, and suggestion from the ANNIE named entity recognition system. The first part of Table 13 shows the experimental results in which two types of news are used for training and the third type for testing. The second part presents the results using 10-fold cross-validation on the three types of documents separately and on the mixed dataset of all three, respectively. The last part shows the experiments using the CoNLL-2003 dataset as training and half of the news dataset as testing. In this part, there are no results for the last three types of entities, as they were not covered in the CONLL'2003 corpus.

Table 14: Experiments on the news dataset using different types of documents as training and test set, respectively: F_1 on the each type of entity is reported. The NLP features are as above, without Named Entity Recognition information suggestions from ANNIE.

	Person	Location	Organisation	Date	Money	Percent
Two types of news for training, another type for test						
UK news	0.902	0.862	0.784	0.883	0.968	0.991
Int. news	0.889	0.895	0.848	0.925	0.919	0.940
Business news	0.905	0.876	0.835	0.817	0.757	0.978
10-fold CV on every of three types and on the mixed, respectively						
Business news	0.942	0.881	0.883	0.874	0.940	0.980
Int. news	0.920	0.920	0.838	0.908	0.868	0.796
UK news	0.898	0.856	0.822	0.861	0.973	0.822
Mixed data	0.908	0.894	0.848	0.872	0.889	0.978
Different training set, the second half of News for test						
CONLL'2003	0.460	0.784	0.734	-	-	-
CONLL'2003 + first half of News	0.871	0.914	0.831	-	-	-
First half of News only	0.914	0.902	0.824	-	-	-

We were also interested in the results without the named entity recognition information. Table 14 presents the results corresponding to those in Table 13 but without ANNIE's named entity suggestions used as a feature. We can see that the result became worse without that feature. However, it seems that there is no significant difference between the results with and without named entity recogniser in many cases.

4.2.2 Template Filling

The results on the seminar corpus are available for quite a few systems. Those include rule learning system such as SRV [Fre00], Whisk [Sod99], Rapier [Cal98], BWI [FK00], SNoW [RY01] and LP^2 [Cir01], as well as statistical learning systems such as HMM [FM99] and maximum entropy [CN02a]. See Section 6 for more details about the previous work.

One problem with carrying out comparisons on the seminar corpus is that the different system used different experimental setups. The SRV, SNoW and maximum entropy systems reported results averaged over 5 runs. In each run the dataset was randomly divided into two partitions of equal size. One partition was used for training and another for test. Furthermore, a third of the training set, randomly selected, was set aside for validation. WHISK's results were from 10-fold cross validation on a randomly selected set of 100 texts. Rapier's and LP^2's results were averaged over 10 runs, in each of which the dataset was randomly split approximately into two halves, one part for training and another part for testing. BWI's and HMM results were obtained via standard cross validation.

The GATE-SVM results reported here are the average over ten runs, following the methodology of Rapier and LP^2. Table 15 presents the results of our system on seminar announcements, together with the results of the other systems. As far as it was possible, we used the same features as by the other systems to enable a more informative comparison. In particular, the results listed in Table 15, including our system, did not use any gazetteer information and named entity recogniser output. The features GATE-SVM used are words, capitalisation information, token types, lemmas, and POS tags. We computed the F_1 measure for each slot. The best results on each slot appear in bold font.

Table 15: Comparison with other systems on CMU seminar corpus: F_1 on each slot. Quadratic kernel and the uneven margin parameter $\tau = 0.4$ were used in the SVM.

	Speaker	Location	Start time	End time
GATE-SVM	0.690	0.813	0.948	0.927
$(LP)^2$	**0.776**	0.750	0.990	0.955
SNoW	0.738	0.752	**0.996**	**0.963**
MaxEnt	0.653	0.823	**0.996**	0.945
BWI	0.677	0.767	**0.996**	0.939
HMM	0.711	**0.839**	0.991	0.595
Rapier	0.531	0.734	0.959	0.946
Whisk	0.183	0.666	0.926	0.861
SRV	0.563	0.722	0.985	0.779

We can see that the results of GATE-SVM are not significantly different from the best results on most slots.

If the information from the ANNIE gazetteer and named entity recogniser is used as additional features, then the micro-averaged F_1 for GATE-SVM is 0.862, which is better than the 0.857 for LP^2 using the same features (see [Cir01]), but is still worse than the 0.872 for the maximum entropy system (see [CN02a]). However, note that our system just used the general NLP features while [CN02a] used genre-specific features (see Section 6 for some details). Furthermore, we did not optimise the parameter settings in the experiments specifically for this corpus (see the discussions about optimising the experimental settings for the CoNLL-2003 dataset above).

For the **jobs postings corpus**, our system was compared to two rule learning systems, Rapier and $(LP)^2$, which were evaluated on this dataset (see [Cal98] and [Cir01] respectively).

Again, in order to make the comparison as informative as possible, we adopted the same settings in our experiments as those used by $(LP)^2$ [Cir03]. In particular, we executed ten runs using a random half of the corpus for training and the rest for testing. The results presented here are the mean of those obtained in the ten runs. In contrast, Rapier's results were obtained via 10-fold cross validation over the entire dataset. Again, only basic NLP features are used: word, capitalisation information, token types, and lemmas.

Table 16: Comparison with other systems on the jobs corpus: F_1 on individual type of entity and the overall figure. Quadratic kernel and the uneven margin parameter $\tau = 0.4$ were used. The highest score on each slot appears in bold.

Slot	**GATE-SVM**	$(LP)^2$	Rapier	Slot	**GATE-SVM**	$(LP)^2$	Rapier
Id	0.977	**1.000**	0.975	Platform	0.801	**0.805**	0.725
Title	**0.496**	0.439	0.405	Application	0.702	**0.784**	0.693
Company	**0.772**	0.719	0.700	Area	0.468	**0.537**	0.424
Salary	**0.865**	0.628	0.674	Req-years-e	**0.808**	0.688	0.672
Recruiter	0.784	**0.806**	0.684	Des-years-e	0.819	0.604	**0.875**
State	**0.928**	0.847	0.902	Req-degree	**0.875**	0.847	0.815
City	**0.955**	0.930	0.904	Des-degree	0.592	0.651	**0.722**
Country	**0.962**	0.810	0.932	Post date	0.992	**0.995**	**0.995**
Language	0.869	**0.910**	0.818	Macro-averaged F_1	**0.808**± 0.063	0.772	0.760

Table 16 presents the results of our system as well as the results of the other two systems on the jobs corpus. GATE-SVM achieves the best results among all three on eight out of the 17 slots and the second best results on nine of the seventeen slots. Overall, the macro-averaged F_1 of GATE-SVM is better than the other systems. However, the significance boundaries indicate that the three system are not significantly different from each other.

4.3 Adaptive information extraction

In adaptive information extraction we are concerned with the ability of an information extraction system to adapt to a new domain or application with minimum effort. From the point of the learning algorithm's view, an adaptive IE is required to learn an initial model from a small number of training examples. Then the performance of system would improve gradually as more and more training instances become available (e.g., from the user annotating new texts).

In order to evaluate our system in an adaptive IE scenario, we evaluated the learning algorithm on a growing number of examples. For both the seminar and jobs corpora, we fist sorted documents in alphabetic order by file name. Then in each experiment the second half of corpus was used as a test set and a small number of documents were picked randomly from the first half for training. For the CoNLL-2003 dataset the training documents were chosen randomly from the training set and the results are reported on the development set. In order to factor out randomness of results, the mean of five runs is reported. The SVM used quadratic kernel and the following features: lemma, case information, tokenkind, gazetteer and ANNIE named entity type.

Table 17 presents the experimental results for different numbers of training documents on the three datasets. We can see that the system performance improved consistently as more training documents were used. In addition, the uneven margin parameter with value less than 1 gave better results, in particular on a small number of training documents.

Table 18 shows that some types of entities can be learned faster than others, due to their more fixed

Table 17: Different numbers of documents for training: macro-averaged F_1 on three datasets. The quadratic kernel was used and results for two different values of the uneven margin parameter are compared.

Dataset	10	20	30	40	50	60	70
	$\tau = 0.4$:						
Seminar	0.555	0.677	0.704	0.734	0.754	0.770	0.787
Job	0.434	0.509	0.539	0.575	0.597	0.608	0.607
CoNLL03	0.606	0.664	0.704	0.722	0.728	0.752	0.764
	$\tau = 1$:						
Seminar	0.377	0.485	0.572	0.621	0.633	0.683	0.701
Job	0.404	0.460	0.496	0.504	0.553	0.575	0.560
CoNLL03	0.462	0.586	0.652	0.683	0.686	0.714	0.735

Table 18: Different numbers of documents for training: macro-averaged F_1 on seminars dataset for every entity types. The quadratic kernel was used. The uneven marginal parameter $\tau = 0.4$.

	10	20	30	40	50	60	70
stime	0.786	0.866	0.866	0.868	0.858	0.870	0.873
etime	0.699	0.839	0.873	0.882	0.881	0.875	0.877
speaker	0.035	0.448	0.540	0.558	0.584	0.628	0.633
location	0.280	0.555	0.535	0.629	0.692	0.708	0.752

internal structure. For example, start and end times can be learned from as little as 10 documents, while at least 60 documents are required to reach similar performance on speaker and location. When interpreting these results one must bear in mind that most documents in the seminars dataset provide only one, or maximum two, examples of each slot (the ratio between number of documents and number of examples per slot in the corpus ranges between 0.9 and 2). Therefore, in this case learning after 10 documents is almost equivalent to learning from 10 to 15 examples per slot.

Table 19: Different number of documents used for training: F_1 measure, quadratic kernel, and uneven margin parameter $\tau = 0.4$.

	Number of training docs	$(LP)^2$	GATE-SVM
stime	30	0.840	0.866
etime	20	0.823	0.839
location	30	0.700	0.535
speaker	25	0.506	0.476

Another system which carried out adaptive IE experiments on the seminars dataset is $(LP)^2$ [CW03]. Table 19 compares our results with those of $(LP)^2$. In a nutshell, our system is better than $(LP)^2$ on etime and stime categories but is worse on location and speaker. Note that the F_1 on speaker for our system increased significantly to 0.539 if only five more training documents were added. However, we do cannot compare this to the results of $(LP)^2$ with 30 training documents, as the paper [CW03] does not provide this information.

In order to evaluate the learning algorithm in an adaptive IE setting on the named entity recognition task, we carried out a number of experiments on the news corpus. When interpreting the results, one must bear in mind that documents annotated with named entities tend to contain a much bigger number of instances per document than the seminars dataset. The average number of examples of named entity types per document ranges between 1.4 for less frequent ones like percentages and 10.7 for the most frequent ones (locations). In comparison, the ratios for the seminar corpus are with the range 0.9 and 2. Table 20 shows the results with all features including gazetteers and named entities from ANNIE, Table 21 – with only gazetteers, and Table 22 – only using word, capitalisation, token kind, and lemma features.

As can be seen from these tables, the results improve substantially by using gazetteers and named entity information. They rise from an average of 0.650 after training on 10 random documents from the whole news dataset, to 0.741 with gazetteers, and up to 0.797 when ANNIE named entity recognition results are used. However, as the number of examples grows the difference becomes less pronounced. For example, on 70 documents, the average on the who data set is 0.854 (Table 22) for the basic features, 0.887 with added gazetteers, and 0.920 with named entity feature from ANNIE.

The three tables also demonstrate that the difference in performance on different named entity types is influenced also by the type of news documents. For instance, Table 22 shows that the performance on Person and 10 documents is much lower on business news (0.665) than that on international political (0.788) and UK political news (0.713). However, the inverse is true for Percent, where the system achieves 0.9 on business news, decreasing to 0.791 for UK political and 0.614 for international political news. Most probably this is due to the fact that different types of named entities appear with different frequencies in the three types of news texts (see Table 12).

5 Experiments with hierarchical classification of entities

The ACE’04 corpus [ACE04] contains 218 news documents in mixed case and 220 other news documents in lower case. The documents in mixed case were from newswire and newspapers, while the documents in lower case were from radio or television news (processed by a speech recognition system). All document were annotated with named entities, which form a 2-level hierarchy. The top level consists of four types, PER, ORG, LOC and GPE. The ORG was divided into 5 sub-categories, the LOC into 8 sub-categories, and the GPE into 6 sub-categories. Table 23 shows the entity types and the number of examples of each every type in the mixed case part of the corpus.

We conducted experiments separately on the mixed case and lower case parts of the corpus. In all these experiments, we used 4-fold cross-validation and the same parameter values as those on the CoNLL-2003 dataset, i.e., window size 4, quadratic kernel, uneven margin parameter 0.5, the $1/j$ weighting scheme for feature vectors from surrounding words, and the probability post-processing

Table 20: Different numbers of documents for training: macro-averaged F_1 on news dataset for every document type and the three mixed and for every entity types. All the NLP features produced by GATE were used.

	10	20	30	40	50	60	70
	Business news						
Person	0.906	0.918	0.919	0.921	0.939	0.937	0.944
Location	0.878	0.884	0.877	0.902	0.890	0.904	0.915
Organisation	0.840	0.856	0.869	0.882	0.889	0.892	0.892
Date	0.789	0.822	0.852	0.867	0.885	0.890	0.887
Money	0.885	0.940	0.950	0.961	0.979	0.974	0.970
Percent	0.969	0.972	0.977	0.978	0.979	0.978	0.982
Average	0.878	0.899	0.907	0.919	0.927	0.929	0.932
	Political int. news						
Person	0.845	0.885	0.908	0.911	0.914	0.922	0.926
Location	0.880	0.904	0.926	0.938	0.940	0.947	0.950
Organisation	0.776	0.797	0.835	0.832	0.831	0.859	0.846
Date	0.818	0.858	0.872	0.893	0.900	0.908	0.902
Money	0.633	0.707	0.868	0.881	0.913	0.940	0.928
Percent	0.590	0.700	0.795	0.810	0.873	0.857	0.923
Average	0.757	0.808	0.867	0.877	0.895	0.906	0.913
	Political UK news						
Person	0.845	0.882	0.901	0.899	0.899	0.900	0.906
Location	0.848	0.876	0.903	0.915	0.910	0.924	0.928
Organisation	0.701	0.778	0.808	0.834	0.839	0.839	0.831
Date	0.765	0.814	0.846	0.868	0.875	0.884	0.896
Money	0.729	0.884	0.948	0.976	0.972	0.977	0.972
Percent	0.642	0.779	0.810	0.748	0.857	0.913	0.869
Average	0.755	0.836	0.869	0.873	0.892	0.906	0.900
	the whole News dataset						
Person	0.852	0.886	0.883	0.891	0.899	0.906	0.912
Location	0.869	0.889	0.903	0.907	0.919	0.924	0.924
Organisation	0.753	0.796	0.823	0.833	0.839	0.851	0.867
Date	0.756	0.819	0.842	0.859	0.869	0.885	0.883
Money	0.738	0.857	0.922	0.919	0.934	0.943	0.970
Percent	0.814	0.889	0.962	0.972	0.968	0.969	0.973
Average	0.797	0.856	0.889	0.897	0.904	0.913	0.920

procedure.

Table 24 presents the results on mixed case documents and on the top 4 categories, both with and without the semantic information from gazetteer and the ANNIE named entity recogniser. We can see that the gazetteer and ANNIE were indeed helpful for recognition. Similar improvements are

Table 21: Different numbers of documents for training: macro-averaged F_1 on news dataset for every document type and the three mixed and for every entity types. The named entity recogniser output was not used.

	10	20	30	40	50	60	70
	Business news						
Person	0.888	0.895	0.921	0.921	0.915	0.922	0.939
Location	0.739	0.816	0.831	0.840	0.858	0.857	0.862
Organisation	0.785	0.823	0.846	0.854	0.865	0.875	0.881
Date	0.727	0.808	0.826	0.848	0.863	0.868	0.862
Money	0.783	0.881	0.985	0.912	0.919	0.046	0.942
Percent	0.898	0.976	0.966	0.978	0.979	0.979	0.982
Average	0.803	0.866	0.881	0.892	0.900	0.908	0.911
	Political int. news						
Person	0.814	0.863	0.888	0.895	0.897	0.913	0.917
Location	0.814	0.859	0.882	0.897	0.899	0.908	0.912
Organisation	0.730	0.778	0.799	0.817	0.827	0.819	0.852
Date	0.746	0.821	0.854	0.858	0.877	0.890	0.899
Money	0.471	0.658	0.743	0.791	0.728	0.816	0.836
Percent	0.650	0.751	0.790	0.857	0.892	0.854	0.879
Average	0.704	0.788	0.826	0.852	0.853	0.867	0.882
	PolitUK news						
Person	0.787	0.846	0.867	0.883	0.882	0.883	0.909
Location	0.704	0.802	0.829	0.833	0.853	0.855	0.970
Organisation	0.643	0.745	0.776	0.790	0.795	0.799	0.817
Date	0.632	0.753	0.805	0.826	0.833	0.843	0.847
Money	0.622	0.834	0.828	0.916	0.938	0.960	0.941
Percent	0.714	0.801	0.856	0.819	0.860	0.886	0.885
Average	0.684	0.797	0.827	0.845	0.860	0.871	0.878
	the whole News dataset						
Person	0.802	0.842	0.859	0.870	0.881	0.884	0.893
Location	0.765	0.814	0.832	0.851	0.861	0.869	0.879
Organisation	0.697	0.755	0.787	0.789	0.814	0.818	0.832
Date	0.709	0.784	0.801	0.824	0.835	0.848	0.859
Money	0.670	0.713	0.844	0.850	0.877	0.873	0.889
Percent	0.802	0.802	0.957	0.960	0.967	0.971	0.970
Average	0.741	0.785	0.847	0.857	0.872	0.877	0.887

observed also on the 19 second level categories. Therefore, in the subsequent experiments the features included those from the gazetteer and ANNIE. We can also see from Table 24 that the performance was quite low for the LOC category in comparison to other categories, which may be due to the small number of LOC examples in the corpus. Similarly, the small number of examples for some sub-categories also resulted in very poor performance (see Table 24).

Table 22: Different numbers of documents for training: macro-averaged F_1 on news dataset for every document type and the three mixed and for every entity types. Neither the gazetteer information nor named entity recogniser output were used.

	10	20	30	40	50	60	70
	Business news						
Person	0.665	0.781	0.821	0.844	0.855	0.884	0.881
Location	0.686	0.756	0.790	0.822	0.818	0.843	0.847
Organisation	0.603	0.706	0.745	0.784	0.787	0.817	0.818
Date	0.691	0.766	0.800	0.851	0.832	0.859	0.866
Money	0.785	0.857	0.870	0.913	0.900	0.928	0.939
Percent	0.900	0.969	0.973	0.977	0.976	0.978	0.979
Average	0.722	0.806	0.833	0.867	0.860	0.885	0.888
	Political int. news						
Person	0.788	0.823	0.849	0.855	0.878	0.877	0.880
Location	0.770	0.825	0.859	0.878	0.887	0.897	0.898
Organisation	0.624	0.719	0.743	0.768	0.783	0.797	0.791
Date	0.672	0.764	0.806	0.837	0.844	0.874	0.868
Money	0.521	0.561	0.720	0.751	0.799	0.830	0.837
Percent	0.614	0.787	0.796	0.808	0.866	0.869	0.881
Average	0.665	0.747	0.795	0.816	0.843	0.857	0.859
	Political UK news						
Person	0.713	0.784	0.817	0.832	0.843	0.847	0.847
Location	0.686	0.744	0.803	0.817	0.840	0.848	0.863
Organisation	0.479	0.628	0.734	0.771	0.783	0.794	0.800
Date	0.587	0.708	0.760	0.789	0.802	0.830	0.840
Money	0.534	0.681	0.877	0.922	0.953	0.947	0.962
Percent	0.791	0.775	0.788	0.813	0.851	0.892	0.765
Average	0.632	0.720	0.798	0.824	0.845	0.860	0.846
	the whole News dataset						
Person	0.734	0.777	0.799	0.824	0.830	0.837	0.850
Location	0.683	0.762	0.501	0.820	0.835	0.847	0.860
Organisation	0.410	0.544	0.636	0.657	0.687	0.703	0.726
Date	0.631	0.718	0.767	0.793	0.810	0.838	0.843
Money	0.738	0.809	0.806	0.823	0.837	0.846	0.880
Percent	0.702	0.869	0.946	0.961	0.955	0.963	0.967
Average	0.650	0.747	0.793	0.813	0.825	0.839	0.854

The next step was to exploit the hierarchical structure. For the top categories ORG, LOC and GPE which were divided into some sub-categories, we implemented the so-called *2-level classification* where we first carried out classification for the top category and then classified into the corresponding sub-categories only those words which were regarded as belonging to the top category by the first classifier. Table 25 shows the results for the 2-level classification in comparison to the

Table 23: Numbers of named entities in the ACE corpus

PER	2119	ORG	1304	LOC	109	GPE	1792
		Commercial	776	Address	1	Continent	23
		Educational	41	Celestial	3	County-or-District	39
		Government	209	Land-Region-Natural	11	Nation	772
		Non-Profit	35	Region-International	12	Other	53
		Other	243	Region-Local	9	Population-Center	632
				Region-National	22	State-or-Province	273
				Region-Subnational	32		
				Water-Body	19		

Table 24: Different NLP features: F_1 for every category and the macro averaged F_1. Gaz refers to semantic information from the gazetteer, Entity refers to named entity information from ANNIE, and WCTL is the four features: word, case information, Tokenkind and lemma.

NLP features	PER	ORG	LOC	GPE	MA F_1
WCTL	0.718	0.553	0.286	0.759	0.579
WCTL+Gaz	0.757	0.596	0.296	0.784	0.608
WCTL+Gaz+Entity	0.844	0.642	0.333	0.826	0.661

Table 25: Results for the sub-categories: comparing the flat classification with the 2-level classifications.

ORG	flat	2-level	LOC	flat	2-level	GPE	flat	2-level
Commercial	0.560	0.556	Address	0.000	0.000	Continent	0.520	0.561
Educational	0.096	0.198	Celestial	0.000	0.000	County-or-District	0.125	0.151
Government	0.364	0.354	Land-Region-Nat.	0.000	0.000	Nation	0.813	0.760
Non-Profit	0.000	0.000	Region-Internat.	0.000	0.000	Other	0.381	0.438
Other	0.242	0.258	Region-Local	0.000	0.000	Population-Center	0.681	0.573
			Region-National	0.000	0.000	State-or-Province	0.489	0.446
			Region-Subnat.	0.188	0.251			
			Water-Body	0.299	0.363			
MA F_1	0.252	0.273		0.061	0.077		0.502	0.488

flat classification where every sub-category was classified directly. The 2-level classification was better than the flat classification in many cases. For macro-averaged F_1, the 2-level classification was better than the flat one in two out of three groups.

In the previous experiments we regarded a document as a sequence of words and took the surrounding words for the current word from the sequence. However, the words belonging to a sentence might correlate more closely than the words from different sentences. Hence it looks more promising to take surrounding words for the current word only from the same sentence rather than across different sentences. Since there are sentence tags in the ACE corpus, we conducted experiments to

Table 26: The results for the document with sentence information.

PER	0.844	ORG	0.645	LOC	0.338	GPE	0.827
		Commercial	0.562	Address	0.000	Continent	0.537
		Educational	0.096	Celestial	0.000	County-or-District	0.125
		Government	0.356	Land-Region-Nat.	0.000	Nation	0.814
		Non-Profit	0.000	Region-Internat.	0.000	Other	0.411
		Other	0.232	Region-Local	0.000	Population-Center	0.678
				Region-National	0.000	State-or-Province	0.509
				Region-Subnat.	0.188		
				Water-Body	0.288		

investigate the importance of sentence information. Table 26 shows the results. When compared to the corresponding results in Tables 24 and 25 which did not use the sentence information, we can see that using sentence information improved slightly the performance for most categories.

Table 27: The results for the lower case part of corpus.

PER	0.842	ORG	0.621	LOC	0.347	GPE	0.818
		Commercial	0.547	Address	0.000	Continent	0.543
		Educational	0.089	Celestial	0.000	County-or-District	0.113
		Government	0.349	Land-Region-Natural	0.000	Nation	0.742
		Non-Profit	0.000	Region-International	0.000	Other	0.369
		Other	0.206	Region-Local	0.000	Population-Center	0.644
				Region-National	0.000	State-or-Province	0.346
				Region-Subnational	0.188		
				Water-Body	0.338		

Finally, Table 27 presents the results for the lower case part of ACE corpus. The lower case part had worse performance than the mixed case part. This is due to lower case documents being less informative than mixed case documents.

6 Related work

This section briefly describes previous work on applying machine learning to IE, in particular those systems which were evaluated on the three datasets used in our experiments. We first describe the applications of SVM to IE. Then we look at the other algorithms evaluated on the CoNLL-2003 dataset. Finally, the rule learning and statistical learning IE systems on the seminar announcements and job postings corpora are reviewed.

6.1 SVM-based Systems

The SVM based system in [IK02] trained four SVM classifiers for each named entity type – besides the two SVMs for start and end words like ours, one for middle words, and one for single word entities. They also trained an extra SVM classifier to recognise the words which do not belong to any named entity. [IK02] used a sigmoid function to transfer the SVM output into a probability and then applied the Viterbi algorithm to determine the optimal label sequence for a sentence. The system was evaluated on a Japanese IE corpus. The features used were POS tag, character type and the word itself. They used the neighbouring words with window size 2. Their experiments showed that the SVM based system performed better than both maximum entropy and rule learning systems on the same dataset using the same features. They also showed that quadratic kernel was better than both linear and cubic kernels on their dataset. [IK02] also described an efficient implementation of the SVM with quadratic kernel.

[MMP03] used a lattice-based approach to named entity recognition and employed SVM with cubic kernel to compute transition probabilities in a lattice. They trained an SVM classifier for every possible transition of tags, meaning that they may have a large number of SVM classifiers. They tested the system on the CoNLL2003 dataset using cubic kernel. The features included the word itself, the character 3-grams and 4-grams that compose the word, the word's length, the word's position in sentence, lemma, POS tag, the output of another named entity recogniser, and the maximum likelihood estimate, based on the training data, of the words prior probability of being in each class. They also took into account the features from neighbouring words (The window size 3 was used). It is interesting to note that they did not use the gazetteer list provided by CoNLL-2003 corpus (see a possible explanation in Subsection 4.1). Their result on the CoNLL2003 corpus is comparable to ours (see Table 11). There are some other applications of SVM for bio-named entity recognition (see e.g. [SYKL04]).

6.2 Other Learning Methods Evaluated on the CONLL'2003 corpus

CoNLL2003 is a typical named entity recognition corpus with newswire articles and entity types similar to the earlier MUC-6 and MUC-7 corpora [SAI98]. Sixteen systems participated in the evaluation. All of them were based on statistical learning, except one system which used rule learning as one of four algorithms which were combined as one classifier. The system with the best score was exactly this combined system, based on robust risk minimisation, maximum entropy, transformation based learning and HMMs, respectively (see [FIJZ03]. The features included words, lemmas and POS tags in a window of size 2, word types, information from a gazetteer[10], and outputs of two other named entity classifiers.

Another system only based on maximum entropy obtained slightly worse results (see [CN03]). They used quite a few features, including local features such as word, capitalisation information, token types, POS tag, bi-gram, and suffixes, and the so-called zone features which was related to the structure of document used, a global feature from the whole document, and finally the information from a gazetteer compiled from a variety of sources including the Internet and the

[10] It is not clear if the gazetteer of CoNLL-2003 corpus was included in their gazetteers or not.

CONLL-2003 gazetteer. Note that another two participating systems were also only based on maximum entropy (see [BON03], [CC03]). In particular, the probability discriminant model used in [CC03] was quite similar to the one in [CN03]. The features used in [CC03] included words, word types, capitalisation information, POS tags, and information from a person name gazetteer consisting of first and last names. They did not use the CoNLL-2003 gazetteer either. However, both the scores of these two systems ([BON03] and [CC03]) were significantly worse than the system described in [CN03], which confirms the conjecture that the appropriate features are as important as the learning algorithm.

The SVM based participating system was discussed above (see [MMP03]).

6.3 Learning Systems Evaluated on Template Filling

SRV is a relational learning (or inductive logic programming) algorithm for IE, which deduces a set of rules for one type of information entity from training examples (see [Fre98a]). It checked every text fragments of appropriate size in document in order to identify if the fragment was an information entity or not. [Fre98b, Fre00] tested SRV for IE on three datasets – the CMU seminars corpus, a collection of 600 newswire articles on corporate acquisitions from Reuters and a collection of web pages of university computer science departments. For the seminars corpus, it used the basic NLP features such as the token itself, length of token, token type (e.g., numeric), orthography (e.g., capitalised) and the previous and following tokens. For the web pages collection, [Fre98b] compared the performance with and without HTML features and found that the genre-specific features were quite helpful. On the newswire collection, [Fre98b] compared the the basic features without and with syntactic information (i.e. the POS tags and the syntactic relations) and lexical semantic (from WordNet) and found that, although linguistic features had good effect in some cases, overall they appear to have little effect on SRV's performance.

WHISK [Sod99], another relational learning system for IE, was tested on collections of structured, semi-structured and free-text documents, such as CNN weather domain, seminar announcements, software jobs postings, and news story articles. WHISK's results on the seminars corpus were not as good as SRV's, which may be attributed to the fact that WHISK used less features – only the token and its semantic class.

Rapier is also a rule based learning IE system (see [Cal98]). It was tested on two dataset: software jobs and seminar announcements. The features used included words, POS tags as assigned by Brill's tagger, and semantic classes taken from WordNet. Its results on seminar announcements are better than SRV. [Cal98] also found that POS tags were helpful in some cases such as the speaker and location slots in seminar announcements but the semantics classes had little impact on Rapier's performance.

BWI (Boosted Wrapper Induction) involved learning a wrapper (boundary detector) for an information entity via a boosting procedure (see [FK00]). It was evaluated on several collections such as seminar announcements, software job postings, Reuters articles, and web pages. The features included token, capitalisation information and token types. [FK00] also considered the neighbouring words as context and found, similar to us, that different datasets have a different optimal window size. One should note that for rule based learning algorithms the training time increases exponentially with window size.

LP^2 is also a rule learning algorithm for IE (see e.g. [Cir01]). In [CW03] LP^2 was tested on three datasets: seminar announcements, software job postings, and a collection of 103 web pages describing computer science courses. It compared three different sets of features. LP^2 used features such as word, lemma, capitalisation information and lexical category. LP^2_{NG} used only the word and capitalisation information and LP^2_G used the rich set of NLP features such as word, lemma, capitalisation, lexical category, semantic types from gazetteer and the outputs of named entity recognition system. [Cir01] also discussed the different effects of window size on different entity types.

[RY01] presented another relational learning based IE system, SNoW. It learned rules via a multi-class classifier by looking at a target fragment and its left and right windows. It was evaluated on the seminar announcements dataset. The features used included words, POS tags and the length information of the fragment.

[FM99] exploited a general statistical model, Hidden Markov Models (HMMs), for IE. It also used the shrinkage technique to deal with data spareness for HMM parameter estimations. It was tested on two corpora, the seminar announcements, and a collection of newswire articles from Reuters. It used similar experimental settings to SRV and obtained better results on the seminars corpus.

[CN02a] used a probabilistic discriminant model for IE and used maximum entropy for parameter estimations. It was tested on several corpora including seminar announcements, the CoNLL-2003 corpus (see [CN03]) and the datasets from MUC-6 and MUC-7 (see [CN02b]). They used quite a few features which were extracted from a local window of the current word as well as from the whole document. They also used some genre-specific features such as the so-called zone related features which are dependent on the structure of documents. For seminar announcements, they also used time expressions which would be particularly useful for recognising the stime and etime slots.

All previous work used features from a window surrounding the current word, as well as features of the word itself. [FK00] and [Cir01] investigated the effect of window size on the performance of rule-based learning and noticed that the computation time increased exponentially as the window size grew. On the other hand, the computation time in an SVM based system only increases linearly with window size. Hence it is easier for the SVM algorithm to select and use the optimal window size. It also should be noted that previous systems treated the features from words in the window as equally important. In other words, this is equivalent to using the equal weighting scheme defined in Section 2. However, our experiments demonstrated that the reciprocal $1/j$ weighting achieves better results (see Section 4.

Basically the rule learning IE systems did not do any post-processing other than simple consistency checking – they treated each type of entity separately. The statistical learning algorithms compute a probability for each entity (or transfer the output into a probability as in the SVM based IE algorithms), such that they can select the best label for a fragment of text based on these probabilities. In order to select the best labels for a sentence, a Viterbi-like search algorithm was usually employed as a post-processor in the statistical learning systems.

[Cal98] and [Cir01] also investigated the effects of growing quantities of training data, which is useful for adaptive IE. [Cal98] also considered active learning, where the system learns an initial model from a small pool of annotated examples and then, based on the learned model, selects additional examples for training.

7 Conclusions

This paper presents an SVM-based algorithm for IE and experiments on several benchmark datasets. The results showed that our system is comparable to other state-of-the-art systems on both traditional IE and adaptive IE tasks. In comparison to other similar SVM-based algorithms, our algorithm is simpler, i.e., it needs a smaller number of SVM classifiers per category than the other two systems discussed respectively in [IK02] and [MMP03]. GATE-SVM also obtained better results than the SVM-based system in [MMP03] on the CoNLL-2003 corpus.

Our algorithm uses an uneven margin parameter, which we showed to be particularly useful for adaptive information extraction on a small amount of training data.

We also investigated two weighting schemes for the features of the surrounding words and showed that the reciprocal weighting scheme performed better than the commonly used equally weighting. We also investigated three post-processing procedures: from using the SVM outputs for begin and end tags separately to selecting the highest probability label based on the output of all SVM classifiers. We found that the probability scheme gave best results.

We also carried out experiments using various combinations of features in order to systematically investigate the effects of different NLP features on the performance of the learning algorithm. The NLP features produced by the general purpose modules from GATE were used and none of them are domain specific, unlike some used by other systems. Words provided most information and basic linguistic features such as capitalisation and lemmas were often quite helpful, however part-of-speech information was not useful. Both the general ANNIE gazetteers and rule-based named entity recogniser were helpful. Despite the fact that GATE provides document formatting information, e.g., HTML tags, such text-specific features were not utilised in our current experiments.

The main reason for applying our machine learning algorithms to corpora like CMU seminars and jobs, instead of an ontology annotated corpus, comes from the lack of corpora for IE, annotated with rich ontological information. Other related projects, e.g., DOT.KOM[11], have also used the seminars and jobs corpora instead.

This work goes one step further by using the ACE corpus which provides hierarchically structured named entities and is thus closest in spirit to the data needed for ontology-based IE. Therefore, our first experiments on using statistical methods for ontology-based IE used this corpus.

A still ongoing recent effort is the Pascal Challenge on evaluating machine learning for information extraction[12]. As part of this initiative, an annotated corpus of workshop calls for papers (CFPs) has been produced. However, similar to the jobs postings, this corpus uses a very flat ontology of approximately 16 classes.

Future work on ontology-based IE in SEKT will fill this gap by using the newly created onto-

[11] http://nlp.shef.ac.uk/dot.kom/resources.html

[12] http://nlp.shef.ac.uk/pascal/

logically annotated corpus (see D2.5.1 Evaluation Tools and Corpora). Another strand of future work on OBIE will be concerned with the creation of user-friendly tools for training the system., possibly augmented with information from the Internet, e.g., the Pankow system [CHS04].

References

[ACE04] ACE. *Annotation Guidelines for Entity Detection and Tracking (EDT)*, Feb 2004. Available at http://www.ldc.upenn.edu/Projects/ACE/.

[BON03] Oliver Bender, Franz Josef Och, and Hermann Ney. Maximum entropy models for named entity recognition. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 148–151. Edmonton, Canada, 2003.

[BSW99] D. Bikel, R. Schwartz, and R.M. Weischedel. An Algorithm that Learns What's in a Name. *Machine Learning, Special Issue on Natural Language Learning*, 34(1-3), Feb. 1999.

[Cal98] M. E. Califf. *Relational Learning Techniques for Natural Language Information Extraction*. PhD thesis, University of Texas at Austin, 1998.

[CC03] James R. Curran and Stephen Clark. Language independent ner using a maximum entropy tagger. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 164–167. Edmonton, Canada, 2003.

[CHS04] P. Cimiano, S. Handschuh, and S. Staab. Towards the Self-Annotating Web. In *Proceedings of WWW'04*, 2004.

[Cir01] F. Ciravegna. $(\mathrm{LP})^2$, 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.

[Cir03] F. Ciravegna. $(\mathrm{LP})^2$, Rule Induction for Information Extraction Using Linguistic Constraints. Technical Report CS-03-07, Department of Computer Science, University of Sheffield, Sheffield, September 2003.

[CMB$^+$02] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, and C. Ursu. *The GATE User Guide*. `http://gate.ac.uk/`, 2002.

[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.

[CN02a] 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.

[CN02b] H. L. Chieu and H. T. Ng. Named entity recognition: A maximum entropy approach using global information. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02)*, Taipei, Taiwan, 2002.

[CN03] H. L. Chieu and H. T. Ng. Named entity recognition with a maximum entropy approach. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 160–163. Edmonton, Canada, 2003.

[Cun05] H. Cunningham. Information Extraction, Automatic. *Encyclopedia of Language and Linguistics, 2nd Edition*, 2005.

[CW03] 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.

[DAH+97] D. Day, J. Aberdeen, L. Hirschman, R. Kozierok, P. Robinson, and M. Vilain. Mixed-Initiative Development of Language Processing Systems. In *Proceedings of the 5th Conference on Applied Natural Language Processing (ANLP-97)*, 1997.

[DH97] A. C. Davison and D. V. Hinkley. *Bootstrap Methods And Their Application*. Cambridge University Press, Cambridge, UK, 1997.

[FIJZ03] R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. Named Entity Recognition through Classifier Combination. In *Proceedings of CoNLL-2003*, pages 168–171. Edmonton, Canada, 2003.

[FK00] Dayne Freitag and Nicholas Kushmerick. Boosted Wrapper Induction. In *Proceedings of AAAI 2000*, 2000.

[FM99] 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.

[Fre98a] D. Freitag. Information extraction from html: Application of a general learning approach. *Proceedings of the Fifteenth Conference on Artificial Intelligence AAAI-98*, pages 517–523, 1998.

[Fre98b] D. Freitag. *Machine Learning for Information Extraction in Informal Domains*. PhD thesis, Carnegie Mellon University, 1998.

[Fre00] D. Freitag. Machine Learning for Information Extraction in Informal Domains. *Machine Learning*, 39(2/3):169–202, 2000.

[IK02] H. Isozaki and H. Kazawa. Efficient Support Vector Classifiers for Named Entity Recognition. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02)*, pages 390–396, Taipei, Taiwan, 2002.

[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.

[LYRL04] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5(Apr):361–397, 2004.

[MMP03] J. Mayfield, P. McNamee, and C. Piatko. Named Entity Recognition Using Hundreds of Thousands of Features. In *Proceedings of CoNLL-2003*, pages 184–187. Edmonton, Canada, 2003.

[MTU+01] D. Maynard, V. Tablan, C. Ursu, H. Cunningham, and Y. Wilks. Named Entity Recognition from Diverse Text Types. In *Recent Advances in Natural Language Processing 2001 Conference*, pages 257–274, Tzigov Chark, Bulgaria, 2001.

[PKK+04] B. Popov, A. Kiryakov, A. Kirilov, D. Manov, D. Ognyanoff, and M. Goranov. KIM – Semantic Annotation Platform. *Natural Language Engineering*, 2004.

[RY01] 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.

[SAI98] SAIC. Proceedings of the Seventh Message Understanding Conference (MUC-7). `http://www.itl.nist.gov/iaui/894.02/-related_projects/muc/index.html`, 1998.

[SM03] E. F. Sang and F. D. Meulder. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada, 2003.

[Sod99] S. Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1):233–272, 1999.

[SYKL04] Y. Song, E. Yi, E. Kim, and G. G. Lee. POSBIOTM-NER: A Machine Learning Approach for Bio-Named Entity Recognition. In *Workshop on a critical assessment of text mining methods in molecular biology*, Granada, Spain (`http://www.pdg.cnb.uam.es/BioLINK/workshop_BioCreative_04/`), 2004.

[YL99] Y. Yang and X. Liu. A Re-Examination of Text Categorization Methods. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 42–49, 1999.

Appendix I - Using the GATE SVM Light Wrapper

This section provides a brief overview of the SVM Light Wrapper in GATE which was used for these experiments. This is by no means meant to be a complete user guide on GATE and its machine learning features, which are documented extensively in [CMB$^+$02]. In order to use the wrapper, one needs to download the latest version of GATE, available from `http://gate.ac.uk`.

We developed a machine learning wrapper (implemented in the java class gate.creole.ml.svmlight.SVMLightWrapper), which gives access to the SVM Light support vector machine software. All the functionality of SVM Light can be accessed, except that there is no support for ranking support vector machines. Information about SVM Light can be found at `http://svmlight.joachims.org/`. SVM Light is written in C, and takes the form of command line programs, rather than a code library. Consequently it is not interfaced directly with GATE, but is instead called as an external process. Data is passed to and from SVM Light using temporary files, but these processes are hidden from the GATE user, so that it appears to be seamlessly integrated. However, SVM Light is not distributed as part of the standard GATE distribution, instead it must be obtained from the website given above, and the two executables, svm_learn and svm_classify placed somewhere on the user's path (that is those directories that the operating system searches for executable files).

The wrapper allows support vector machines to be created which either do boolean classification or regression (estimation of numeric parameters), and so the class attribute can be boolean or numeric. Additionally, when learning a classifier, SVM Light supports transduction, whereby additional examples can be presented during training which do not have the value of the class attribute marked. Presenting such examples can, in some circumstances, greatly improve the performance of the classifier, so a facility was added to allow this technique to be used from within GATE. The class attribute can be a three value nominal, in which case the first value specified for that nominal in the configuration file will be interpreted as true, the second as false and the third as unknown. Transduction will be used with any instances for which this attribute is set to the unknown value. It is also possible to use a two value nominal as the class attribute, in which case it will simply be interpreted as true or false.

The other attributes can be boolean, numeric or nominal, or any combination of these. If an attribute is nominal, each value of that attribute maps to a separate SVM Light feature. Each of these SVM Light features will be given the value 1 when the nominal attribute has the corresponding value, and will be omitted otherwise. If the value of the nominal is not specified in the configuration file or there is no value for an instance, then no feature will be added. Like MAXENT models, SVM Light models are not updateable, and so are created and trained the first time a classification is attempted. However, the SVM Light Wrapper supports the batch classification mode of the machine learning processing resource. If <BATCH-MODE-CLASSIFICATION/> is specified in the <ENGINE> part of the configuration file[13], then all the instances for a document will be passed to the wrapper at one time, rather than them being passed one at a time. Using this option will result in a great improvement in efficiency in most circumstances.

[13] For information on configuration files for the machine learning modules, please consult the Gate User Guide, available from `http:`
`gate.ac.uk`

The SVM Light wrapper allows both data sets and models to be loaded and saved to files in the same formats as those used by SVM Light when it is run from the command line. When a model is saved, a file will be created which contains information about the state of the SVM Light Wrapper, and which is needed to restore it when the model is loaded again. This file does not, however, contain any information about the SVM Light model itself. If an SVM Light model exists at the time of saving, and that model is up to date with respect to the current state of the training data, then it will be saved as a separate file, with the same name as the file containing information about the state of the wrapper, but with '.NativePart' appended to the filename. These files are in the standard SVM Light model format, and can be used with SVM Light when it is run from the command line. When a model is reloaded by GATE, both of these files must be available, and in the same directory, otherwise an error will result. However, if an up to date trained model does not exist at the time the model is saved, then only one file will be created upon saving, and only that file is required when the model is reloaded. So long as at least one training instance exists, it is possible to bring the model up to date at any point simply by classifying one or more instances (i.e. running the model with the training parameter set to false).

Options for the SVM Light Wrapper

CLASSIFIER-OPTIONS: These are the same options as those passed to svm_learn on the command line, and should be specified in the same format. The only difference is that the user should not specify whether regression or classification is to be used, as the wrapper will detect this automatically, based on the type of the class attribute, and set the option accordingly.