An Adaptive Information Extraction System based on Wrapper Induction with POS Tagging

Rinaldo Lima
Centro de Informática, UFPE
50740-540 Cidade Universitária,
Recife, PE, Brazil
rjl4@cin.ufpe.br

Bernard Espinasse
LSIS UMR CNRS 6168
Domaine Universitaire de St Jerôme,
F-13997, Marseille Cedex 20, France
bernard.espinasse@lsis.org

Fred Freitas
Centro de Informática, UFPE
50740-540 Cidade Universitária,
Recife, PE, Brazil
fred@cin.ufpe.br

ABSTRACT
Information Extraction (IE) performs two important tasks: identifying certain pieces of information from documents and storing them for future use. This work proposes an adaptive IE system based on Boosted Wrapper Induction (BWI), a supervised wrapper induction algorithm. However, some authors have shown that boosting techniques face difficulties during the processing of natural language texts. This fact became the rationale for coupling Parts-of-Speech tagging with the BWI algorithm in our proposed system. In order to evaluate its performance, several experiments were carried out on three standard *corpora*. The results obtained suggest that the union of POS tagging and BWI offers a small gain of 3-5% of performance over the original BWI algorithm for unstructured texts. These results position our system among the very best similar IE systems endowed with POS tagging, according to a comparison presented and discussed in the article.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: *Information gathering and extraction* - I.2.6 [**Learning**]: *Wrapper Induction*.

General Terms
Experimentations

Keywords
Information extraction, wrapper induction, boosting, supervised classification, POS tagging, machine learning.

1. INTRODUCTION
The Information Extraction (EI) field has the goal of extracting and integrating information data in a collection of documents (corpus) of a same domain, as well as, of reducing textual information to tabular structures of easier manipulation [7].

As a result, many researches have been conducted in the development of IE systems increasingly adaptable to several domains [14] and different types of texts, ranging from web pages created by databases to *Call for Papers* written in natural language. These adaptive IE systems [7] use machine learning techniques for wrapper induction. In the context of EI, a wrapper is a program that can extract information from a corpus. Wrapper induction uses ML algorithms to generate extraction rules from a set of documents previously annotated, rather than have them manually defined by a knowledge engineer.

[3] propose an adaptive IE algorithm based on supervised classification for wrapper induction. This wrapper induction technique, Boosted Wrapper Induction (BWI), induces wrappers intended to be applied on semi-structured documents (HTML/XHTML), and achieves good performance on extraction from more or less structured texts.

The goal of our research is to propose an adaptive IE system from web pages that takes into consideration morphosyntactic structures of natural language in which these pages are written. The system we propose, named WEPAIES (Web Pages Adaptive Information Extraction System), is a modular system specialized on IE from web pages. After preprocessing web pages, in special POS tagging, the IE task is based on supervised wrapper induction by using BWI techniques.

In the section 2, we present the basic concepts of the adaptive IE field, and the BWI technique as well as POS tagging which allows specify morphosyntactic structures of natural language. Section 3 presents our system WEPAIES, a modular adaptive IE system based on supervised wrapper induction. The influence on performance in IE tasks obtained by taking into account POS tagging in conjunction with BWI on the reference corpora are discussed in Section 4. In the Section 5, we compare our system performance with others existents IE systems by experiments using 3 reference corpora.

2. BOOSTED WRAPPER INDUCTION AND POS TAGGING
2.1 Boosted Wrapper Induction (BWI)
Usually, induced wrappers are not suitable to process natural language texts because each induced rule (contextual pattern) presents a low recall. However, [3] addressed this problem by using *boosting* (a general technique for improving the accuracy of a weak learning algorithm) to learn many such patterns and combining their results. As a result, the set of learned rules present both high precision and high recall.

The Boosted Wrapper Induction (BWI) algorithm [3, 6] implements the previous approach for inducing simple contextual patterns (wrappers) for finding the start and the end (boundaries) of a field to extract. In BWI algorithm, a wrapper W = <*F, A, H(k)*> consists of a set of F (*fore*) and A (*aft*) *detectors* - patterns that detect the start and the end of a target field; and a *length function H(k)* which estimates the prior probability that a field has

SAC'10, March 22-26, 2010, Sierre, Switzerland.

length k. BWI estimates these probabilities by constructing a frequency histogram H(k) recording the number of fields of length k occurring in the training set. More information about BWI as well as experiments and results analysis can be found in [3, 6, 7].

2.2 POS (Part-Of-Speech) Tagging

However, only inducted wrappers are clearly not enough for natural language texts, ought to their low precision. Relevant information in these texts can be identified by some regularity present in their linguistic patterns. These patterns often refer to Parts of Speech (POS) from the clauses. Because certain facts are usually expressed by certain part of speech, POS (Parts-of-Speech) tagging linguistic analysis proved to be useful for determining parts of speech of tokens that can be identified by IE systems as classification features or elements of extraction rules.

3. PRESENTATION OF WEPAIES

WEPAIES (Web Pages Adaptive Information Extraction System) is a modular extraction system that takes into account the morphosyntactic structure of the web pages being processed. In WEPAIES information extraction is performed by wrappers induced with the BWI technique. A distinguishing feature of the system lies on the use of POS tagging in order to improve recall. In the following sections we describe the general WEPAIES architecture and various modules that compose it.

3.1 System Architecture

The WEPAIES system is composed of various modules (see Figure 1), each addressing a specific processing task. Before performing the actual information extraction, a preparation phase of web pages is necessary. This preparation phase consists of the tasks of Cleaning (Web Cleaner pages), Tokenization, Feature Extraction, and POS tagging. Several software modules ensure these various tasks, which will be explained in the next section.

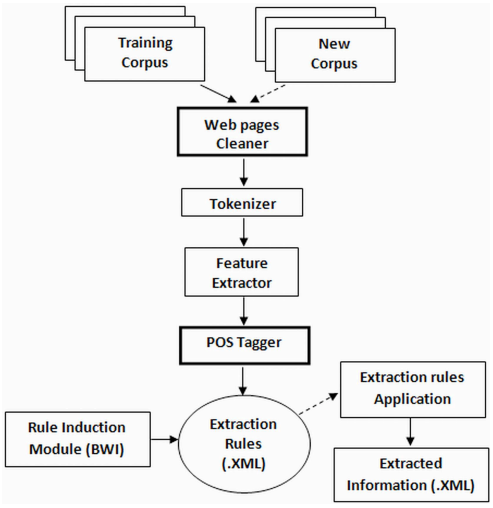

Figure 1. WEPAIES functional architecture.

WEPAIES is in fact a software architecture integrating, for each task, and after customizations, software modules already developed and available. Thus, for the cleaning task of web pages, WEPAIES integrates HTMLCleaner tool [5]; for POS tagging, the QTAG tool [13]; and finally for the supervised wrapper induction with BWI techniques, an adapted version of TIES developed at IRST in Trento [9]. Figure 1 illustrates the functional architecture of WEPAIES.

Thus, WEPAIES system automatically learns extraction rules from a corpus of web pages, which are first annotated (by the user) with a predefined set of XML tags that identify the positive examples. Next, POS tags taking into account the morphosyntactic structures of the web pages are added. Finally, the generated extraction rules will be used to automatically extract information from new unseen web pages.

3.2 Preprocessing Phase modules

As already mentioned, the preprocessing of web pages for the induction of extraction rules or their application in extraction is necessary. This is achieved by preprocessing modules for cleaning, tokenization, feature extraction and POS tagging.

Web Page Cleaner. WEPAIES requires well-formed XHTML documents as input and this has conducted us to customize and use HTMLCleaner [5], a tool for web page cleaning and transformation. This tool corrects missing HTML tags and removes irrelevant tags and text parts.

Tokenization. This phase identifies tokens: elementary parts of natural language words, symbols, punctuation marks, etc. The resulting sequence of tokens is converted into the TIES Input Format (TIESIF) which describes tokens with their attributes.

Feature Extraction. The tokens originated from the previous step are defined in terms of 12 attributes [9] which are the results of discrete-value functions. Each token is described by a set of attributes thus producing a 12-dimensional vector.

POS Tagger. One goal of this work is to integrate, into the original TIES architecture, a module concerned with POS tagging in order to assess its influence on the extraction results of the learning component of the BWI algorithm when using non-structured documents as input. For that, the stochastic POS tagger QTAG [13] was used. It creates a set of labels (tags), lexical and contextual probabilities from a corpus manually annotated.

By default, after the Tokenization/Feature Extraction phases, TIES generates a file without taking morphosyntactic annotations into consideration. Our idea is to insert the POS tags of the tokens proposed by QTAG into this file. Thus, the induction module can finally use an enriched tokenized corpus with POS information.

3.3 Rule Induction Module

As already stated, this task is performed by an adapted version of the TIES system developed at IRST in Trento [9] which implements the BWI algorithm (section 2.1). In this section we limit ourselves to explain how the induced wrappers and information extracted are expressed by the components of learning and extraction, respectively. For more details about the learning and extraction components of the BWI algorithm, we refer the reader to [3, 9].

3.3.1 Model generation: Rules Learned

WEPAIES induced rules are expressed by wrappers formed by a prefix, p, followed by the information to be extracted and ending with a suffix, s. Indeed, a rule describes a couple of patterns <p, s> that surrounds the field to be extracted. Induced wrappers are stored in XML format. The fragment below shows an induced wrapper for the <speaker> slot of the Seminars corpus.

```
<wrapper label="speaker">
<fore-detector>
<detector>
       <pattern type="prefix">
         <feature name="token" value="Who"/>
         <feature name="single_char_token" value="true"/>
       </pattern>
       <pattern type="suffix">
          <feature name="alpha_token" value="true"/>
       </pattern>
</detector>
...
</wrapper>
```

This rule means that a *speaker* can be found just after the token "Who" followed by a single character token and just before any alphanumeric token.

3.3.2 Model Application: Extraction

The wrapper model (the set of induced rules) in this phase is applied during the extraction step. Entities extracted from a new corpus are again stored in XML format. The following fragment shows the tokens that constitute the target slot <speaker> and their positions in documents provided by the tags <entity> and its attribute pair (*start*, *end*).

```
<entity-list>
  <entity name="speaker" src=".\CMUAN~3G.DER"
        start="142"  end="151">
      <token start="142" len="3"> Mr.</token>
      <token start="146" len="5"> Okada</token>
  </entity>
     ...
</entity-list>
```

We tested our system against standard corpora. The experiments and results are discussed in the next section.

4. EXPERIMENTS AND RESULTS

Firstly, this section presents the 3 standard corpora against which WEPAIES has been evaluated. Then we report the experimental results in order to assess the gain that POS tagging can provide for WEPAIES performance.

4.1 Selected Corpora

The first corpus experimented was the Seminars corpus, which consists of a collection of 485 conference announcements. For each document in the corpus, we had to extract 4 targets slots: *location*, *speaker*, *stime* and *etime*.

The 2nd corpus was the Jobs Corpus. It contains job offers in the field of Computer Science and is composed of 300 documents. Each of them can contain up to 17 target slots (Table 1) - with information about employers, jobs requirements, etc. [1].

Table 1. Slot frequency of Job corpus.

Jobs	Platf	Lang	Area	City	State	App
	709	851	1005	659	452	590
	Title	Recruit	Post-d	Country	Salary	Req-y-e
	457	312	302	345	141	166
	Comp	Des-y-e	Req-deg	Des-deg	Id	
	298	43	83	21	304	

The 3rd corpus contains 1,100 documents describing 850 Workshop Call for Papers (CFP) and 250 Conference CFP, established during the competition for IE systems (Pascal Challenge) [8]. The large majority of the documents are from the Computer Science field. Each Workshop document can contain 8 slots, while Conferences can have up to 3 slots (see Table 2).

Table 2. Slot frequency of CFP corpus.

Slot	Corpus Frequency			
	Training	%	Test	%
workname	543	11.8	245	10.8
workacro	566	12.3	243	10.7
workhome	367	8.0	215	9.5
workloca	457	10.0	224	9.9
workdate	586	12.8	326	14.3
workpape	590	12.9	316	13.9
worknoti	391	8.5	190	8.4
workcame	355	7.7	163	7.2
confname	204	4.5	90	4.0
confacro	420	9.2	187	8.2
confhome	104	2.3	75	3.3
Total	**4583**	**100**	**2274**	**100**

4.2 POS Tagging Results and Discussion

The experiments carried out in this section examine the influence of POS tagging on the extraction results of all 3 corpora. For all experiments in this work, we have fixed the number of tokens for the fore and after detectors (the so-called *lookahead* parameter *L*) as 3, following the suggestions from [3, 6] in order to have a good trade-off between performance and running time. In the results below, performance results are reported using standard IE measures of *Precision* (*P*), *Recall* (*R*) and *F-Measure* (*F1*). The experiments on Seminars and Jobs corpora were conducted using 10-fold cross-validation, and 4-fold cross-validation on the CFP corpus. Figures 2, 3, 4 and 5 show the results.

Slot	P	R	F1	P	R	F1
stime	0.985	0.979	0.982	0.984	0.983	0.983
etime	0.989	0.969	0.979	0.988	0.974	0.981
location	0.961	0.912	0.936	0.953	0.924	0.938
speaker	0.962	0.944	0.953	0.960	0.965	0.962
	(a) **no POS**			(b) **with POS**		

Figure 2. Results obtained on Seminars.

Discussions of the CFP corpus results

As shown in Figure 4, POS tagging provided a slight increase on the F-measure for the majority of slots. For instance, the *confacro* slot presented gain of more than 5%. The lowest result of the algorithm considering all slots individually was that of confhome slot, ought to the low representativeness of this slot in the corpus. Furthermore, we found out that the module responsible for the tokenization of documents does not recognize email addresses as an entity. A possible way of improving the results for this type of slot would be either allowing the tokenization phase to recognize it or increasing the window size. However, even with the lack of email recognition, the results with POS tagging improved the BWI in circa 2%.

Discussion of results on all corpora

Figure 5 presents the overall results in order to appreciate POS tagging gains on all corpora. We observed that for the CFP corpus, we obtained the better results concerning POS tagging. On the other hand, in the corpora of Seminars and Jobs (Figures 2 and 3) there was practically no difference. These results can be justified by analyzing the more structured nature of the documents in these corpora. Indeed, the wrappers induced can have very good performance without using a larger hypotheses space, i.e., with no need for POS information. We also noted that these results are similar to those shown by [12]. The authors have carried out the same experiments we have just discussed about the Seminars and Jobs corpora using the SVM algorithm and they also obtained a tiny gain on the Seminars corpus and even a negative impact of POS tagging on the Jobs corpus. More

precisely, the performance loss was less than 1% for the latter. Among the 17 slots that constitute the extraction template for the Jobs corpus, we could note that more than half of them are presented in a regularly structured form. These facts lead us to the conclusion that the use of POS tagging combined with BWI pays off against highly unstructured texts.

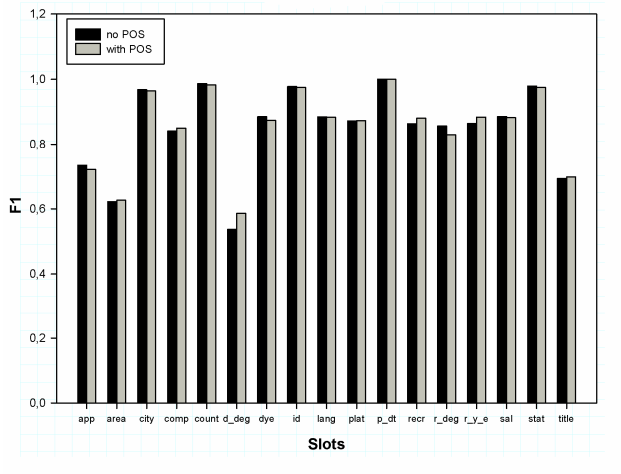

Figure 3. Results obtained on Jobs corpus.

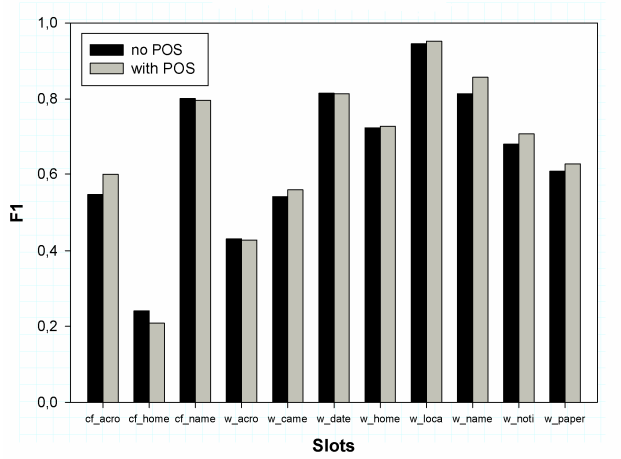

Figure 4. Results obtained on CPF corpus.

Corpus	P	R	F1	P	R	F1
Seminars	0.974	0.953	0.963	0.971	0.964	0.967
Jobs	0.945	0.778	0.853	0.939	0.780	0.853
CFP	0.891	0.571	0.696	0.896	0.591	0.712
	(a) no POS			(b) with POS		

Figure 5. Influence of POS tagging on corpora.

5. COMPARATIVE EVALUATION

First, we define the evaluation criteria used for achieving more accurate and reliable comparisons. Then, the selected IE systems are briefly described and finally we discuss the results.

5.1 Brief Description of Compared Systems

Rapier [1] is an IE system which aims to extract information from free text. Its learning algorithm incorporates techniques of inductive logic programming and learns patterns that are not constrained by a fixed window size but includes constraints on words and on POS tags surrounding the data to extract.

GATE-SVM [12] uses the SVM algorithm for supervised token classification. This system employs a variant of SVM with uneven margins [11] which has better generalization performance than the original SVM.

Yaoyong [8] is the GATE-SVM predecessor and their classifiers use a right/left context window of 10 tokens (20 in total).

SIE (Simple Information Extraction) [4] is based on supervised learning (SVM) and employs an interesting technique for token filtering (Instance Pruning).

Amilcare LP^2 [2] is a rule induction system based on the supervised algorithm (LP)^2 using LazyNLP [2] which generalizes extraction rules beyond the flat word structure, while keeping a good performance in highly structured texts. This system was the champion of the Pascal Challenge Competition [8].

5.2 Results and Discussion

5.2.1 Comparisons on Seminars and Jobs corpora

The definition of an evaluation methodology and the availability of standard annotated corpora do not guarantee that experiments carried out with different approaches and algorithms can be compared in a reliable way [10]. We adopted, as much as possible, the same evaluation methodology, by following the recommendations proposed by [10] (definition of corpus partitions, preprocessing tasks and results reported).

In this light, comparisons on Seminars and Jobs corpora become problematic. Note also that all systems employ POS tagging with the exception of SIE. However SIE was chosen because we took the best performance systems in the Pascal Challenge contest.

The experimental setup data and the performance results presented on Tables 3 to 6 were obtained from [4, 12, 2].

In the tables in this section, the Context column indicates the values of the lookahead *L* (section 4.2) and the window size *W* parameters which determine the context in number of tokens observed by the selected IE learning algorithms.

Table 3. Experimental setup on Seminars corpus.

System	Evaluation method	Used features	Context
(LP)^2	Random split (50/50) - 10 times	word, capitalization and POS	W=5
GATE-SVM	Random split (50/50) - 10 times	word, capitalization, token type, lemma and POS	W=5
Rapier	Random split (50/50) - 10 times	word, POS and Wordnet	-
SIE	2-fold cross validation - 5 times	word, capitalization, lemma, alphanum. and punctuation	W=10
WEPAIES	Random split (50/50) - 10 times	word, capitalization and POS	L=3

Table 4. Performance (*F1*) by slot on Seminars corpus.

	speaker	location	stime	etime	All slots
SIE	-	-	-	-	86.6
GATE-SVM	69.0	81.3	94.8	92.7	86.2
(LP)^2	77.6	75.0	**99.0**	95.5	86.0
Rapier	53.0	72.7	93.4	96.2	77.3
WEPAIES	**86.2**	**88.8**	93.9	**96.7**	**91.4**

Table 5. Experimental setup on Jobs corpus.

System	Evaluation method	Used features	Context
(LP)^2	Random split (50/50) – 10 times	word, capitalization and POS	W=5
GATE-SVM	Random split (50/50) - 10 times	word, capitalization, token type, lemma, POS, NER and gazetteer	W=3
Rapier	10-fold cross validation	word, POS and Wordnet	-
WEPAIES	Random split (50/50) - 10 times	word, capitalization and POS	L=3

Table 6. Performance (F1) by slot on Jobs corpus.

Slot	$(LP)^2$	GATE_SVM	Rapier	WEPAIES
id	**100.0**	97.7	97.5	98.1
title	43.9	49.6	40.5	**67.4**
company	71.9	77.2	69.5	**78.9**
salary	62.8	86.5	67.4	**89.2**
recruiter	80.6	78.4	68.4	**86.1**
state	86.7	92.8	90.2	**96.9**
city	93.0	95.5	90.4	**96.5**
country	81.0	96.2	93.2	**98.8**
language	**91.0**	86.9	80.6	88.5
plataform	80.5	80.1	72.5	**86.9**
application	**78.4**	70.2	69.3	73.1
area	**66.9**	46.8	42.4	51.6
req_y_exp	68.8	80.8	67.1	**86.4**
des_y_exp	60.4	81.9	87.5	**89.9**
req_degree	84.7	**87.5**	81.5	78.6
des_degree	**65.1**	59.2	72.2	47.6
post date	99.5	99.2	99.5	**100.0**
All-slots	**84.1**	**80.8**	**75.1**	**83.8**

Discussion of results on Seminars Corpus

Table 4 results demonstrated that WEPAIES was clearly superior on three slots while SIE, SVM-GATE and $(LP)^2$ achieved a similar overall performance.

It is worth mentioning that GATE-SVM system used a richer feature set than other systems [12] in this experiment. With its full feature set, WEPAIES achieved a performance of 96.7% (F-measure). Similarly $(LP)^2$ obtained 89.7% on this corpus using NER and gazetteers. There were no detailed scores for SIE, which explains the missing values in the Table 4.

The good score for speaker and location slots confirms the adequacy of the BWI algorithm on more structured documents. For the BWI algorithm, if a target slot is simply preceded or followed by a set of tokens represented by wildcard characters available in its hypotheses space, prefix and suffix detectors can easily learn this context. That is the case here where the slots are often preceded by identification tags (e.g. "Speaker: Dr. X"), or followed by information easily identifiable. While other rule-based IE methods are primarily designed to identify contexts *outside* target slots, the BWI algorithm learns patterns that occur *inside* the target slots [6].

The current version of the WEPAIES tokenizer module is optimized to identify, as early as possible, instances of dates, times and more common abbreviations. This could explain the better result for etime slot. On the other hand, for stime slot, it seems that it is necessary more context to achieve a good score. The more focused NLP approach of $(LP)^2$ algorithm obtained the best score for this slot.

To sum up, with this experiment, we have drawn to the conclusion that in the presence of more structured corpora, like the Seminars, the BWI algorithm is enough to achieve the best performance for most of the slots.

Discussion of results on Jobs corpus

In Table 6, all-slots scores of all systems are in micro F-measure, except for GATE-SVM which uses the macro F-measure [12, 2]. In general, all systems demonstrated uniform performance on this corpus. WEPAIES has achieved the highest scores in 11 of 17 slots, while $(LP)^2$ was superior in 6 slots. However, these performance differences are very small. Other slots such as id and post-date are highly regular, which explains the superior performance of all systems on these slots.

For WEPAIES, the largest positive difference of performance was noted for the title slot. On the opposite, the biggest negative difference was noted for the des-degree slot. By analyzing the annotations for the former, we see that it has a highly variable size and its content is more important than its context to identify it. On the other hand, the lowest representativeness of the latter (only 21) explains this relatively low score.

The bottom line is that statistical significance tests demonstrated that the systems compared on this corpus are not considerably different from each other.

5.2.2 Comparison on CFP corpus

Experimental setup

Table 7 shows the experimental setup used for the comparative evaluation. Results of the compared systems in this section can be found in [8]. We used 4-fold cross-validation for all systems.

Table 7. Experimental setup for the comparative evaluation.

System	Used features	Context
$(LP)^2$	word, capitalization, token type, lemma, POS, NER, gazetteer	w = 5
SIE	word, capitalization, token type, lemma, NER, gazetteer	w = 10
Yaoyong	word, capitalization, token type, lemma, NER , gazetteer	w = 10
WEPAIES	word, capitalization, token type, POS, entities (date and time)	L = 3

CPF Results and Discussion

All systems presented a wide variation regarding the ability to identify certain slots (Table 8). Amilcare $(LP)^2$ achieved the best F-measure scores in 6 of 11 slots, while WEPAIES obtained the best scores on 4 slots.

Table 8. System results (*P*, *R* and *F1*) by slots on CFP corpus.

		WORKSHOP								CONFERENCE		
System	Sc	name	acro	date	home	loca	pape	noti	came	name	acro	home
Amilcare	P	0.656	0.887	0.769	0.864	0.621	0.876	0.889	0.876	0.792	0.922	0.656
$(LP)^2$	R	0.241	0.884	0.632	0.619	0.402	0.851	0.889	0.865	0,422	0.888	0.280
	F	0.352	**0.865**	0.694	0.721	0.488	**0.864**	**0.889**	**0.870**	0.551	**0.905**	**0.393**
Yaoyong	P	0.629	0.738	0.810	0.656	0.611	0.719	0.867	0.764	0.649	0.619	0.368
	R	0.539	0.523	0.666	0.870	0.674	0.763	0.821	0.736	0.411	0.348	0.093
	F	0.580	0.612	0.731	**0.748**	0.641	0.740	0.843	0.750	0.503	0.445	0.149
SIE	P	0.852	0.733	0.850	0.672	0.812	0.841	0.921	0.911	0.795	0.667	0.556
	R	0.539	0.259	0.451	0.419	0.406	0.617	0.795	0.687	0.344	0.235	0.067
	F	0.660	0.383	0.589	0.516	0.542	0.712	0.853	0.783	0.481	0.348	0.119
WEPAIES	P	0.889	0.906	0.918	0.718	0.990	0.906	0.925	0.849	0.953	0.930	0.706
	R	0.825	0.275	0.729	0.735	0.916	0.477	0.569	0.414	0.691	0.443	0.122
	F	**0.856**	0.422	**0.813**	0.726	**0.952**	0.625	0.705	0.556	**0.801**	0.600	0.209

The lowest performance was held for 3 Conference slots, once they have a relatively low frequency in the corpus, which indicates an insufficient number of examples in order to achieve good generalizations.

$(LP)^2$ obtained the lowest scores for workshop name, workshop location and conference name slots, which demonstrates that their techniques do not guarantee a desirable performance on all types of slots. Examining the documents, one can see that these difficult slots for $(LP)^2$ are not specified by their contexts, but rather they are determined by their contents and places in documents. By contrast, WEPAIES displayed a good performance for these slots because the BWI algorithm can learn about regularities that occur inside the fields being extracted. This is probably the reason why our system obtained the best overall precision among the tested IE systems (see Table 9).

The results of Table 9 also stress that the systems have more precision than recall. This is probably a general feature of IE

systems, which generally are intended to perform extraction tasks that assign a higher cost to false positives. In addition, we can only attain high recall at the expense of precision, and vice versa.

Again, the explanation for the lowest WEPAIES score (the conference home page *c-hom* slot) lies on the fact that this slot is the less representative in the entire CFP corpus. Anyway, this slot hampered the performance of all other systems (Table 8). However, in WEPAIES, results could be improved for this slot either allowing the tokenizer module to recognize home pages URLs or increase the numbers of learning examples.

Finally, Table 9 shows that WEPAIES was the most accurate of all systems for the CFP corpus, but it had the lowest recall. To sum up, its F-measure performance was similar to Yaoyong and SIE and lower than the $(LP)^2$ on this corpus.

Table 9. Comparison between 4 systems on CFP corpus.

System	P	R	F1
Amilcare	84.3	70.3	**76.7**
Yaoyong	70.2	**71.7**	70.9
SIE	75.5	65.2	70.0
WEPAIES	**89.6**	59.1	71.2

5.2.3 Conclusions from the Comparative Evaluation

The comparative experiments above showed that WEPAIES is superior to other systems on the more structured Seminars Corpus, and it is comparable on the semi-structured Jobs corpus. In addition, WEPAIES induced wrappers for highly unstructured text corpora tend to be more accurate than other systems while keeping a reasonable recall. However, even with POS tagging, WEPAIES scored lower in terms of F1 than $(LP)^2$ on this type of corpus.

Actually, the BWI algorithm employs a more expressive set of rules due to the use of wildcards that generalize better than LP^2 algorithm on more structured documents. This set of rules contributes positively to the experimental results of the BWI algorithm. BWI induces simple extraction rules which are closely equivalent to LP^2 best rules [2]. Moreover, in BWI, the boosting technique is used to focus on examples in which the learner has low performance in order to create additional rules. Instead, the LP^2 algorithm employs a learning approach based on a simple coverage algorithm [2].

Finally, for free texts in natural language, we can conclude that the superiority of LP^2 algorithm over the others lies in its more advanced NLP approach.

6. CONCLUSIONS

We have presented WEPAIES, an adaptive IE system based on supervised wrapper induction. It takes into account the syntax of natural language by including POS tagging.

Regarding the effective gain that POS tagging can provide for WEPAIES, we have detected an improvement in the scores of almost 5% for some target slots of a highly unstructured natural language corpus. On the other hand, for two more structured corpora, there was almost no gain. In addition, concerning highly regular targets fields, the system achieved a perfect score.

Experiments with comparative evaluation suggest that WEPAIES is superior to other IE systems on more structured text corpus, and it has achieved comparable scores on semi-structured text corpus. In contrast, WEPAIES has achieved a lower performance than $(LP)^2$ on natural language documents.

In general, wrappers produced by WEPAIES achieved higher precision than all other systems while keeping an acceptable recall.

ACKNOWLEDGEMENTS

This work was funded by the LSIS lab under the Click&Go Project. We thank ITC-irst (Centro per la Ricerca Scientifica e Tecnologica, Trento, Italy), which conceded us the license to use TIES. The authors also thank Shereen Albitar and Sébastien Fournier from the LSIS lab for their collaboration, and Alberto Lavelli from ITC-irst for his support on the TIES system usage.

REFERENCES

[1] Califf M. E, Mooney R. J. Relational learning of pattern-match rules for information extraction. In *Proc. of the 16h National Conference on AI (AAAI-99)*, 1999, 328-334.

[2] Ciravegna, F. $(LP)^2$, *Rule Induction for Information Extraction Using Linguistic Constraints*. Technical Report CS-03-07, Dep. of CS, Univ. of Sheffield, Sheffield, 2003.

[3] Freitag D., Kushmerick N. Boosted Wrapper Induction. In *Proc. of the 17h National Conf. on AI (AAAI-2000)*, 2000.

[4] Giuliano C., Lavelli A., Romano L. Simple Information Extraction (SIE): A Portable and Effective IE System. In *Proc. of the EACL-06 Workshop on Adaptive Text Extraction and Mining (ATEM-2006)*, Trento, Italy, 2006.

[5] Girardi, C. HtmlCleaner: Extracting Relevant Text from Web Pages. In *Proc. of WAC3 2007 - 3rd Web as Corpus Workshop*. Louvain-la-Neuve, Belgium, 15-16, 2007.

[6] Kauchak D., Smarr J., Elkan C. *Sources of Success for Information Extraction Methods*, Technical Report CS2002-0696. UC, San Diego, 2002.

[7] Kushmerick, N., Thomas B. *Adaptive Information Extraction: Core Tech. for Information Agents*, Springer, 2003, 79-103.

[8] Ireson N., Ciravegna F., Califf M. E., Freitag D., Kushmerick N., Lavelli A. Evaluating machine learning for information extraction. In *Proc. of the 22nd Int. Conf. on ML*, Vol. 119, Bonn, Germany, 2005, 345 - 352.

[9] TIES. *Trainable Information Extraction System*. Dot.Kom project, 2004. Available at: http://tcc.itc.it/research/textec/tools-resources/ties.html

[10] Lavelli A., Califf M. E, Ciravegna F., Freitag D., Giuliano C., Kushmerick N., Romano L. IE Evaluation: Criticisms and Recommendations. In *Workshop on Adaptive Text Extraction and Mining, AAAI-2004*, 2004.

[11] Li, Y., Shawe-Taylor, J.: The SVM with uneven margins and Chinese document categorization. In *Proc. of the 17th PACLIC,* Singapore, 2003, 216–227.

[12] Li Y., Bontcheva K., Dowman M.; Roberts I., Cunningham, H. *Ontology Based Information Extraction (OBIE) v.1*, SEKT deliverable, University of Sheffield, 2004.

[13] Mason O., Tufis D. Tagging Romanian Texts: a Case Study for QTAG, a Language Independent Probabilistic Tagger. In *Proc. of 1st LREC*, Granada, Spain, 1998, 589-596.

[14] Tang J., Hong M., Zhang D., Liang B., Li, J. *Information Extraction: Methodologies and Applications*. DCS-Tsinghua University, 2007.