2015 IEEE 27th International Conference on Tools with Artificial Intelligence

Relation Extraction from Texts with Symbolic Rules Induced by Inductive Logic Programming

Rinaldo Lima, Fred Freitas
Informatics Center, Federal University of Pernambuco
Recife, Brazil
{rjl4, fred}@cin.ufpe.br

Bernard Espinasse
LSIS, Aix Marseille University
Marseille, France
bernard.espinasse@lsis.org

Abstract— **Relation Extraction (RE) is the task of detecting semantic relations between entities in text. Most of the state-of-the-art RE systems rely on statistical machine learning techniques which usually employ an attribute-value representation of features. Contrarily to this trend, we focus on an alternative approach to RE based on the automatic induction of symbolic extraction rules. We present OntoILPER, an RE system based on Inductive Logic Programming which uses a domain ontology in its extraction process. Several experiments are discussed in this paper over the reACE 2004/2005 reference corpora. The results are encouraging and seem to demonstrate the effectiveness of the proposed solution**.

Keywords— *Relation Extraction; Ontology-based Information Extraction; Ontology Population; Inductive Logic Programming;*

I. Introduction

The goal of IE consists in discovering and structuring information found in semi-structured or unstructured documents, leaving out irrelevant information [10]. One of the most important subtasks in IE is Relation Extraction (RE) which is able to detect and characterize semantic relations among entities in text. In other words, RE aims at determining if a given relation between two entities holds, and then assigning a relation type label to it.

Many of the state-of-the-art RE systems rely on statistical machine learning methods, such as feature-based and tree-kernels-based methods [10] [22] [4]. These statistical learning techniques are usually based on a propositional hypothesis space for representing examples, i.e., they employ an attribute-value representation with robust results. However, this representation is not able to effectively capture structural information from parse trees without loss of information [4]. Another line of research on RE concerns the application of Inductive Logic Programming (ILP) [11] for inducing extraction rules [8][9][17][18]. In fact, several ILP-based RE systems have already been proposed (cf. related work section) with good results, but very few of them have used external resources, e.g., ontologies, as background knowledge in their RE process.

The RE system presented here, *OntoILPER*, subscribes to the main idea of employing ILP for RE, enabling prior domain knowledge to be explicitly integrated during the construction of the classification model. The integration of domain ontologies in OntoILPER IE process has the big advantage of improving system portability by allowing the adaptation of system behaviour via chances in the ontology [20]. OntoILPER can also be regarded as an Ontology-based Information Extraction (OBIE) system, performing the ontology population (OP) task [20].

In previous work [23] [24], we have conducted comparative experiments with state-of-the-art RE systems on four benchmark datasets for RE. The yielded results showed that OntoILPER outperforms some RE systems and is very competitive with the rest.

In this paper, we focus on experiments over the reACE 2004/2005, two reference corpora from the news broadcast domain. We provide analyses on experimental results, presenting a detailed discussion of both quantitative and qualitative aspects of the final learned extraction rules.

The remainder of this paper is organized as follows: Section 2 is dedicated to related work about ILP-based RE systems. Section 3 briefly introduces ILP. Section 4 presents how ILP is used in OntoILPER in order to induce symbolic extraction rules for RE. Section 5 discusses experimental results achieved by OntoILPER on the reACE 2004/2005 reference corpora. We conclude and give some perspective of this work in Section 6.

II. Related Work

In this section, we describe state-of-the-art ILP-based systems performing RE found in the literature, which regard some similarities to ours.

Reference [8] proposes a RE system that takes as input dependency trees as relational structures composed of binary predicates representing the edges of dependency graphs. In their solution, the text preprocessing component is based on the GATE framework [28] and the Stanford parser [12]. Applying the notion of Least General Generalization from [15], they generate a set of rules expressed as non-recursive Horn clauses satisfying some criteria of consistency, e.g., all rules must cover a minimum number of positive examples. Then, the generated rules are used for constructing a binary vector of attributes for each example. Finally, the resultant vectors are used for training a SVM classifier [19]. Their system was evaluated on ACE 2003 dataset and obtained 0.568 (F1-measure).

The RE system proposed in [17] applies typed dependencies and ILP as a learning component for RE. It is based on a less expressive propositional learning technique as the key component for rule induction. The authors reported experimental results concerning only one relation (*located_in*) on a small corpus of 13 Wikipedia pages about birds.

1082-3409/15 $31.00 © 2015 IEEE
DOI 10.1109/ICTAI.2015.40

In [18], the authors developed an ILP-based system suitable to learn rules for extracting information from definitions of geographic entities in text (Slovene language). This system is used as a component in a spatial data recommendation service. The authors focus on the extraction of the five most frequent relations ("isA", "isLocated", "hasPurpose", "isResultOf", and "hasParts") in 1,308 definitions of spatial entities. For that, they choose the classical Progol ILP system [13] which induces Horn clauses.

In [9] was introduced an ILP-based system for the Spatial Role Labeling (SpRL) problem which defines two subtasks: the identification of the words playing a role in the descriptions of spatial concepts; and the classification of the role that these words play in the spatial configuration. The authors employed kLog [6], a framework for kernel-based relational learning that uses graph kernels and can take profit of BK in the form of logic programs. Their system relies on the Charniak Parser [2], for POS tagging and dependency parsing; and LHT [29], an automatic semantic role labeling tool. In order to extract spatial relations, kLog is first employed for extracting relational features which are used as input in a propositionalization step. Finally, a SVM classifier is built from the propositionalized features.

The work presented in this paper differs from these related works in two important aspects. First, considering external resources, such as ontologies, none of the surveyed ILP-based systems above make use of such prior knowledge in their RE process. On the contrary, OntoILPER subscribes to the idea of making domain knowledge explicit via the domain ontology. In OntoILPER, relying on the domain ontology in the RE process enables: (i) reasoning for improving RE; (ii) storage of the extracted information in a more flexible formalism than databases, (iii) performing Ontology Population; and (iv) system portability, by allowing the adaptation of its behaviour via changes in the ontology. Second, from the experimental point of view, none of the previous work carried out substantial experiments using several corpora. Thus, we can draw the conclusion that the use of ILP for RE has not been yet fully exploited by substantial assessments on standard RE corpora.

III. Inductive Logic Programming

The first-order learning technique, Inductive Logic Programming (ILP), is able to generate models from complex data structures, such as graphs or multiple tables. Such models can be used for classifying new examples into positive or negative classes. ILP uses first order clauses as a uniform representation language for examples, background knowledge (BK), and hypotheses [11]. Besides the ability of ILP to deal with structured data and to express knowledge in the powerful language of logic programs for describing the induced hypothesis/patterns, learning can be performed by considering available expert knowledge.

The general approach underlying ILP can be outlined, as formally introduced in [13], as follows.

Given:

- a finite set E of examples, divided into positive E^+ and negative E^- examples, both expressed by non-empty sets of *ground facts* (definite clauses without variables), and
- *background knowledge BK*, consisting of a finite set of extensional (ground) or intentional (with variables) Horn clauses. Horn clauses consist of first-order clauses containing at most one positive literal.

The goal is to induce a *correct hypothesis H* (or a *theory*) composed of first-order clauses such that

- $\forall e \in E^+ : BK \wedge H \models e$ (*H* is *complete*), and
- $\forall e \in E^- : BK \wedge H \not\models e$ (*H* is *consistent*).

In practice, it is not always possible to find a correct hypothesis that strictly attends both criteria above, and therefore, both criteria on BK must be relaxed. Reference [11] presents a more detailed introduction to ILP.

IV. Using ILP for Relation Extraction

In OntoILPER, the RE process is performed in two distinct phases, as illustrated in Fig. 1. First, in the Rule Induction phase, a set of rules is induced by a general ILP system from an annotated learning corpus given as input. Then, in the Application phase, the set of the induced rules (learned in the Rule Induction phase) is applied on the candidate relation instances from an unseen document. This process identifies relations instances that are used for populating the domain ontology. In both phases, a previous preprocessing stage takes place in which several Natural Language Processing (NLP) tools generate rich linguistic annotations followed by the representation of all such annotations as BK.

Next, we briefly describe the main components of OntoILPER shown in Fig. 1.

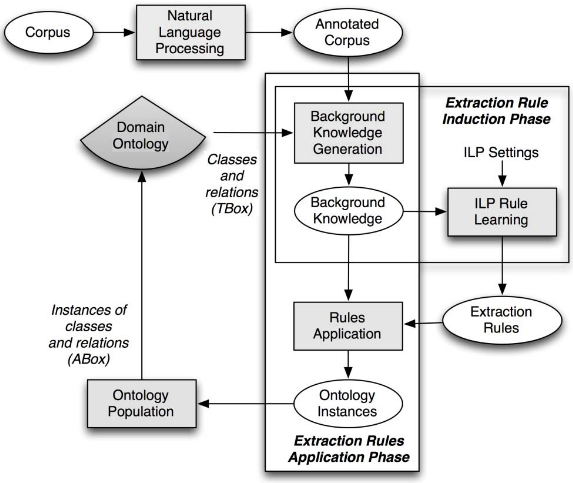
Corpus
Natural Language Processing
Annotated Corpus
Extraction Rule Induction Phase
Domain Ontology
Background Knowledge Generation
Classes and relations (TBox)
ILP Settings
Background Knowledge
ILP Rule Learning
Instances of classes and relations (ABox)
Rules Application
Extraction Rules
Ontology Population
Ontology Instances
Extraction Rules Application Phase

Figure 1. Overview of OntoILPER architecture.

Natural Language Processing. This component automatically annotates the input corpus. The corpus annotation process provides all the morphosyntactic and semantic aspects of the examples which constitute the basis for deriving the rich set of features either in learning or application phase. It essentially deals with texts in English language and integrates the Stanford CoreNLP [30] and OpenNLP tools [31]. These tools are integrated in a pipelined architecture performing the following NLP subtasks on the input corpus: sentence splitting, tokenization, POS tagging, lemmatization (which determines the base form of words), chunking analysis, Named Entity Recognition (NER), and dependency parsing [12].

BK Generation. This component automatically generates and represents relevant features from both the annotated set of documents and the domain ontology. All these annotations about features, the domain ontology, and the examples are converted as BK, which in fact correspond to a Prolog factual base in OntoILPER. We distinguish four main groups of features:

- *Lexical features* concern the word itself, its lemma, length, and general morphological type information.
- *Syntactic features* consist of POS tags of the tokens, head word of nominal, prepositional or verbal chunk. Such features also represent bi-grams and tri-grams of prior and consecutive POS tags of words as they appear in the sentence. In addition, chunking-related features including its type, its head word, and its relative position to the main verb of the sentence, are also provided in this group of feature.
- *Semantic features* include the named entities and any additional entity attributes found in the input corpus. For instance, the entity types such as Person, Location, and Organization are provided by the ACE datasets [26] [27].
- *Structural features* denote all the relational features derived from the graph-based model of sentence representation [24] in OntoILPER. In our sentence representation formalism, the sequencing of token are preserved. Moreover, the grammatical dependencies between two tokens in a sentence define a dependency graph that is combined with the sequencing model mentioned above.

Since Prolog is used as the representation language of examples in OntoILPER, domain entities, relations, and all types of features mentioned earlier are converted to their corresponding Prolog predicates. More detailed about these features and predicates can be found in [23] [24].

ILP Rule Learning. This component is rooted on the GILPS ILP system which provides ProGolem [16], a bottom-up ILP system that has demonstrated competitive performance among others ILP systems. During the learning phase, mode declarations [13] define the *language bias* which delimits and biases the possibly huge hypotheses search space. Other parameters are also provided in order to avoid the overfitting problem of the final induced rules. More information about the language bias and the parameters can be found in [23].

An induced rule for the *part_whole* relation is illustrated next.

```
part_whole(A,B):-  t_gpos(A,nn),  t_next(A,B),
                          t_subtype(B,state-or-province).
```

This rule classifies a relation instance of the *Part-Whole* relation containing two adjacent tokens (or phrases) in a sentence; where the first one (A) is a noun, and the second one (B) is tagged, with respect to the domain ontology, as a relation instance of the “State-or-Provence” subtype class. This rule highlights that places (A) like cities are located, or are part of either a *State* or *Provence*.

Rule Application. During the Application phase, the set of learned extraction rules are passed as input to the Rule Application component that applies them on the knowledge base (Prolog factual base) which was generated from new documents similar to the ones used in the learning phase. As a result of this process, new instances of relations are identified and extracted.

Ontology Population. The extracted instances identified by the Rule Application component are added as new instances into the domain ontology. This process is also known as Ontology Population [20]. In fact, OntoILPER as an OBIE system can be regarded as an OP system, since it is able to assimilate extracted instances into the domain ontology. We rely on the OWL API [32] for implementing the ontology population service.

More information about OntoILPER can be found in [23] [24].

V. Experimental Results and Analysis

This section reports and discusses experimental results on RE achieved by OntoILPER. Our goal is to investigate the effectiveness of OntoILPER according to the following experimental questions (EQ):

- (EQ-1) Which combination of linguistic BK is the best one during the rule learning phase?
- (EQ-2) In an OBIE scenario, what is the influence of ontological hierarchical information on the results?
- (EQ-3) What are the qualitative aspects of the final learned rules?

In what follows, the adopted assessment methodology is presented, including the corpora and the evaluation metrics used. In addition, the answers to the aforementioned EQ’s are provided.

A. Experimental Settings

Datasets for RE. The Automatic Content Extraction (ACE) programme [25] defines the task of Relation Detection and Categorization which aims at detecting and classifying relations between entities according to a predefined ontology. The ACE programme proposed the ACE datasets (2004/2005) for RE which consist of a collection of texts from newswire and broadcast news. Contrarily to previous ACE programmes, these datasets introduced a type and subtype hierarchy to entities and relations mentions, a crucial step towards OBIE [20], which makes the task even more challenging.

In this work, we relied on the revised versions of the ACE 2004/2005 datasets introduced by [7]. These datasets, a.k.a. reACE datasets, are the result of several transformation steps (refactoring, preprocessing, and reannotation) that normalize the two original ACE datasets so that they adhere to a common notion of relation that is more intuitive and simple [7]: a relation instance denotes a predicate over two arguments, where the arguments represent concepts in the real world. Another positive aspect of the transformations is that it facilitates the evaluation and tuning of machine learning algorithms addressing the RE problem; and, for these reasons, they were chosen instead of the original ACE versions.

Tables I and II show the type/subtype relation hierarchy distributions of the reACE 2004/2005 datasets, respectively. The entity types in both reACE datasets were also refactored into 4 types, namely: PER (person), ORG (Organization), GPL (Geo-Political/Location), and FVW (Facility /Vehicle/ Weapon).

Tab. III shows some positive examples of relations found in the reACE 2004 dataset. The negative examples, as required by our ILP-based solution, are artificially generated using the same strategy described in [14].

TABLE I. RELATION DISTRIBUTION IN REACE 2004 DATASET

reACE 2004 - Relation **Type**/Subtype Hierarchy	Freq
Employee-Membership-Subsidiary (EMP ORG)	
Employee-Staff	303
Employee-Executive	220
Member-of-Group	80
General-Affiliation (GEN AFF)	
Located	352
Citizen-Resident-Religion-Ethnic	98
Part-Whole (PRT WHOLE)	
Part-Whole	174
Subsidiary	100
Personal-Social (PER SOC)	
Business	35
Family	15
Total	**1377**

TABLE II. RELATION DISTRIBUTION IN REACE 2005 DATASET

reACE 2005 - Relation **Type**/Subtype Hierarchy	Freq
Organization-Affiliation (ORG AFF)	
Employment	228
Membership	36
General-Affiliation (GEN AFF)	
Located	280
Citizen-Resident-Religion-Ethnic	39
Part-Whole (PRT WHOLE)	
Geographical	119
Subsidiary	47
Personal-Social (PER SOC)	
Business	16
Family	42
Total	**807**

TABLE III. RELATION EXAMPLES OF THE REACE 2004 DATASET

Relations: *type.subtype(arg1, arg2)*	*Example Phrases*
PER-SOC.business(John, superiors)	*John´s superiors ...*
EMP-ORG.emp-exec(Investors, Wall	*Investors on Wall Str...*
EMP-ORG.emp-staff(ABC, John Martin)	*Here's ABC's John M..*
GPE-AFF.citizen/resident(voters, Missouri)	*Some Missouri voters*

Evaluation Metrics and Cross-Validation. The performance evaluation is based on the Information Retrieval classical measures, i.e., Precision *P*, Recall *R*, and *F1*-measure [1]. In all experiments reported here, we used 5-fold cross-validation that provides unbiased performance estimates of the OntoILPER learning phase.

Theory Compression Ratio. The theory compression ratio (TCR) is defined as the measure of the generalization degree of the set of the learned rules (a theory), also denoting their qualitative aspect. This measure is defined by (1):

$$TCR = \frac{\text{number of rules in the learned theory}}{\text{number of positive examples in the training set}} \quad (1)$$

Note that, since each rule must cover at least one positive example, the number of learned rules for a given training set of examples will always be less than or equal to the total number of positive examples. Therefore, the TCR will always resides between 0 and 1, with a lower value indicating a more general theory; whereas a score close to 1 means that the set of rules has just "memorized" the examples (overfiting problem). Moreover, the TCR is a well-motivated and meaningful measure of the quality of a learned theory because it is rather general and objective, since it consists of an unbiased measurement of how many non-overlapping rules would be required to cover every positive example of the training dataset.

B. Evaluation on Features

This experiment aims at analysing the results on relation subtypes (9 in total) by gradually incorporating groups of features during learning. Tab. IV summarizes the results of the combinations of the features including structural and attributive predicates that correspond to nine feature subspaces of interest (Lines 1-9 in Tab. IV). The first feature subspace (*Line 1*) constitutes the *baseline* in our benchmark, i.e., the smallest feature subspace with only morphological features plus the structural predicate *next/*2 that links a token to its successor in a sentence. The other feature combinations are further divided into three categories: structural features (in italics), attributive features (normal font), semantic and NER features (in bold).

The performance improves as more features are used, starting with the F-measure of 53.40% and reaching 81.80% for the reACE 2004 dataset. In the reACE 2005, the best overall F1 performance (71.86%) may indicate that this dataset is more difficult than the reACE 2004. Actually, this is due to the fact that, in the reACE 2005 dataset, some relations (particularly *bussiness*) are very poorly represented with only 16 positive examples, which hampers the overall score. OntoILPER was not able to induce any rule for the *business* relation in all the assessed combinations.

TABLE IV. CONTRIBUTION OF DIFFERENT FEATURES OVER RELATION SUBTYPES IN THE REACE 2004/2005 DATASETS

		reACE 2004			**reACE 2005**		
ID	**Features**	**P**	**R**	**F1**	**P**	**R**	**F1**
1	Baseline	81.09	39.81	53.40	60.53	25.12	35.52
2	+*C*	80.17	47.13	59.36	75.05	34.03	46.80
3	+*D*	81.01	46.93	59.43	72.91	36.51	48.65
4	+*D*+*C*	89.01	54.40	67.53	74.81	38.14	50.48
5	+*D*+*C*+P	91.16	62.04	73.83	81.75	44.24	57.37
6	+*D*+*C*+P+Cr	93.30	66.68	77.77	83.68	50.43	62.91
7	+*D*+*C*+P+Cr**+N**	93.04	67.12	77.99	80.59	51.39	62.68
8	+*D*+*C*+P+Cr+**A**	92.20	71.13	80.31	83.03	63.38	**71.86**
9	+*D*+*C*+P+Cr**+A+N**	92.91	73.07	**81.80**	82.30	61.85	70.62

Baseline = Morphological + Next, C = Nominal and verbal chunkings,
D = Dependencies, P = POS tagging, Cr = Chunking-related features,
N = NER, A = reACE Corpus types.

Discussion. *Lines* 2-4 in Tab. IV demonstrate the usefulness of structural features in the generated models. Thus, the system achieved more than 14% of improvement in F1-measure when comparing the baseline (*Line 1*) with the other structural predicates (*Line 4*), i.e., chunks and dependencies features. Taken separately (*Lines 2 and 3*), chunk and dependencies features practically brought the same boost in performance in the reACE 2004 dataset, which was around 6%. On the other hand, for the reACE 2005 dataset, chunk information was even more effective, as it increases F1 score by 11.28% against only almost 5% due to dependency information.

The above results agree with previous work [22][10] in which the authors reported that, for the original ACE 2004/2005 datasets, chunk or syntactic parsing tree information reflects the more effective structural features for RE. Inspecting the reACE 2004/2005 datasets, we found that most of the candidate entities are encapsulated by nominal chunks, which indicates the quite local characteristic of the semantic relations in the sentences of the reACE corpora. Actually, the distance distribution (in number of tokens) between two related entity instances was analyzed for both datasets. The results revealed that there exists about 67% (in reACE 2004) and 60% (in reACE 2005) of relation instances in which their arguments are separated by at most two tokens. Such results also suggest that each feature subspace alone already captures most of the useful structural information between tokens for RE in these experiments. Due to the locality of semantic relations in reACE 2004/2005, more complex features like dependency trees can only take effect in the remaining minority of long-distance relations. Furthermore, as previously demonstrated in [4], the full parsing of sentences, e.g., dependency parsing, is more susceptible to parsing errors than chunking analysis. Consequently, for such kinds of short-distance relations, sequencing information may be even more reliable than syntactic or dependency information.

Note that the number of chosen features has a direct impact on the size of training examples, since more features are added to the background knowledge, which would certainly require more computational resources. As a result, depending on either the domain or the application, one should take into account this trade-off when selecting the features for RE.

Incorporating both POS and chunking-related features (*Lines 5 and 6* from Tab. IV, respectively) also contribute to performance improvement. Particularly, POS information (*Line 5*) increases the F1 score by more than 6.0 units on both datasets.

The last three lines of Tab. IV display the contribution of attaching the semantic type information to the extraction models. Concerning the impact of the semantic-related features, such as NER and ACE entity types and subtypes in *Line 7-9*, ones can notice that, these features lead to significant performance increases in both datasets.

For the reACE 2004 dataset, a tiny improvement (0.22%) was achieved by using named entities as semantic features (compare *Line 6* with *Line 7*). Surprisingly, these same entities had a reverse effect, decreasing the final F1 score in 0.23 points for the reACE 2005. A closer look at the results revealed that the applied NER component conflicts with the entity types provided by the annotations in the reACE 2005 dataset, generating more false positives.

To conclude, these tests suggest that more accurate semantic information about entity instances can contribute a great deal for RE. This is not surprising, given that semantic information, e.g., classes and subclasses from an ontology, typically impose strong constraints on the types of the entities participating in a relation, indicating that such kind of feature has an crucial discriminative power in RE.

C. Experiments on Hierarchical Classification

As already mentioned, reACE 2004/2005 corpora defined both entity and relation hierarchies, providing a useful scenario for evaluating OntoILPER as an OBIE system, since hierarchical classification is made possible using these datasets.

Accordingly, we evaluated OntoILPER in order to assess its effectiveness when it takes into account the relation taxonomical information from the input domain ontology. The relation hierarchy with three levels is described next, from the most specific level to the more general one [22]:

- *subtype classification* at leaf level, consisting of 9 relations in reACE 2004, and 8 relations in reACE 2005, respectively;
- *type classification*, which denotes 4 middle level relation for both corpora;
- *relation detection*, which denotes the classification task of predicting if a relation holds between two entity instances. This last task can also be considered as a simple binary classification task.

The experiments were conducted separately on the first two classification levels listed above. In each experiment on hierarchical classification, the model is trained and tested on the corresponding level of the relation labels using the complete feature set available in OntoILPER.

Tables V, VI, and VII summarize the results of the subtype and type classification. Besides *P, R,* and *F1*, Tab. VI and VII display the number of positive examples (*E+*) by type, the number of rules in the final theory (*#Rules*), and its theory compression ratio (*TCR*). The zero result for the *business* relation in reACE 2005 is due to very few instances available for training.

Discussion. From a broad view, the average results shown in Tables V, VI, and VII reveal that it is more difficult to classify on deeper levels of the relation hierarchy for both corpora. The reACE 2005 dataset was the one that most profit of the hierarchical classification on a shallow level. Indeed, comparing the average results reported on Tab. V (left part) with those in Tab. VI, it is clear that the *type* classification on reACE 2004 obtained a significant improvement compared to its *subtype* classification.

The analogous comparison between subtype classification (Tab. V right part) and type classification (Tab. VII) over the reACE 2005 shows that the overall improvement was even more substantial.

On the other hand, a comparison of type/subtype classification results put in evidence that:

TABLE V. PERFORMANCE RESULTS OF RELATION SUBTYPES ON BOTH DATASETS

	reACE 2004				reACE 2005			
Rel. Type	**Rel. Subtype**	**P**	**R**	**F1**	**Rel. Subtype**	**P**	**R**	**F1**
EMP_ORG	Employ-Staff	78.10	86.90	82.27	Employ	89.60	86.22	87.88
	Employ-Exec	95.49	77.00	85.25	-	-	-	-
	Member	92.18	76.82	83.80	Member	94.30	71.03	81.03
GEN_AFF	Citizen-Resident	98.81	69.58	81.66	Citizen-Resident	100.00	61.10	75.85
	Located	83.28	80.09	81.65	Located	86.00	84.10	85.04
PERS_SOC	Business	100.00	69.42	81.95	Business	0.00	0.00	0.00
	Family	100.00	39.11	56.23	Family	92.70	57.70	71.13
PRT_WHL	Part-Whole	93.20	83.38	88.02	Geo	100.00	62.10	76.62
	Subsidiary	95.10	75.30	84.05	Subsidiary	95.80	72.51	82.54
	Avg	**92.91**	**73.07**	**81.80**	**Avg**	**82.30**	**61.85**	**70.62**

TABLE VI. CLASSIFICATION RESULTS OF RELATION TYPES ON THE REACE 2004

Rel. Type	**#E$^+$**	**#Rules**	**TCR**	**P**	**R**	**F1**
EMP_ORG	603	65	0.11	86.00	84.00	84.99
GEN_AFF	450	51	0.11	86.90	78.90	82.71
PER_SOC	50	18	0.36	100.00	64.40	78.35
PRT_WHL	274	33	0.12	91.00	81.60	86.04
Total	**1377**	**167**				
Avg			**0.18**	**90.98**	**77.23**	**83.54**

TABLE VII. CLASSIFICATION RESULTS OF RELATION TYPES ON THE REACE 2005

Rel. Type	**#E+**	**#Rules**	**TCR**	**P**	**R**	**F1**
ORG_AFF	264	38	0.14	88.70	77.80	82.89
GEN_AFF	319	60	0.19	94.40	70.40	80.65
PER_SOC	58	19	0.33	100.00	58.30	73.66
PRT_WHL	166	35	0.21	87.60	72.30	79.22
Total	**807**	**152**				
Avg			**0.22**	**92.68**	**69.70**	**79.56**

- The performance scores for the PER_SOC on the type classification level of both corpora rank best among the four relations types. Moreover, the type model yielded the highest possible precision on both corpora, with also a boost on recall.
- The GEN_AFF type classification achieves a small improvement in recall for both corpora.
- EMP_ORG and PART_WHL type classification produced comparable results with their corresponding subtype classification on reACE 2004 corpus.
- ORG_AFF and PART_WHL relations in reACE 2005 did not benefit of the more abstract level of type classification.

These results are again in accordance with the previous ones reported in [21] and [22], as they reveal that is more difficult to classify on deeper levels of the hierarchy because there are less examples per class, since the classes become more similar as the classification level gets deeper. The authors also argued that the closer distance among the classes at subtype level generally causes the performance decreasing, as this can make the classifiers at deeper levels more unstable.

Theory Compression Ratio. In Tables VI and VII, further results in terms of TCR are provided.

Comparing the overall result in terms of TCR on both corpora, one draws to the conclusion that more rules were necessary to cover the examples of the reACE 2005 dataset, with an average TCR of 0.22, against 0.18 on the reACE 2004. Yet, these TCR scores are also reflected in the overall F-measure, significantly lower for the reACE 2005 dataset.

For the three relations types that both datasets have in common, i.e., GEN_AFF, PER_SOC and PRT_WHL, the number of final rules in their respective models are approximately equal, particularly for the last two relation types. However, considering the different proportion of examples of the PRT_WHL type relation in both corpora, this relation was responsible for the major gap in the TCR score in approximately 0.9 units.

The bottom line with respect to the theory ratio assessments is that the proposed solution generates theories of tractable size in the experiments on both datasets.

D. *Qualitative Analysis of the Induced Rules*

One of the main advantages of the learned models in OntoILPER, is that they are expressed in symbolic form, enabling the user to further inspect the most important characteristics of the learned models. Note that this would not be possible using statistical classifiers that only return a numerical value denoting the likelihood of an example belonging to a given class. Next, we discuss about (i) the rule size distribution, and (ii) the contribution of feature types in the final set of rules.

Discussion on Rule Size. Fig. 2 displays the rule size distribution in the induced set of extraction rules.

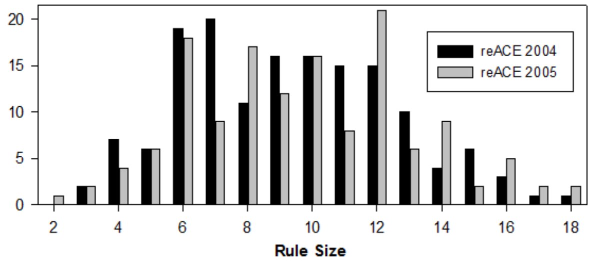

Figure 2. Distribution of rule sizes in reACE 2004/2005 datasets.

For both corpora, one can see that the great majority of learned rules have less than 12 ground atoms in their body part, which corresponds to 80% of the learned rules. This leads to other desired qualities of these rules: they are easier to be understood by a human domain expert. Another interesting finding revealed by Fig**.** 3 concerning both datasets is that, the shorter the final rules, the better the level of rule generalization of the training examples. In other words, the rule size distribution can inform how general or, contrarily, how specific the final rules are.

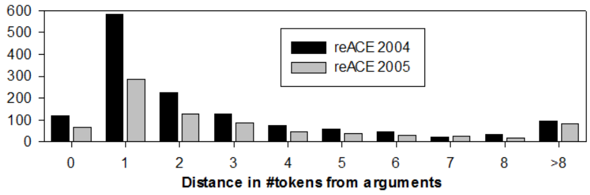

Figure 3. Distribution of the number words between the arguments in relations.

Discussion on Feature Contributions. The objective here is to investigate the contribution of different features (ground atoms) on the set of rules. Fig. 4 displays the contribution of five groups of features (in percentage points). Not surprisingly, due to the same domain and source of both datasets, each group of features shows similar contribution level. Yet, the structural and semantic groups of features were predominant, as these two groups of features were responsible for almost 60% of participation in the set of rules. Interestingly, this confirms the usefulness of the proposed graph-based model for sentence representation in OntoILPER [23] [24] that puts in evidence the relational and semantic aspects of the examples.

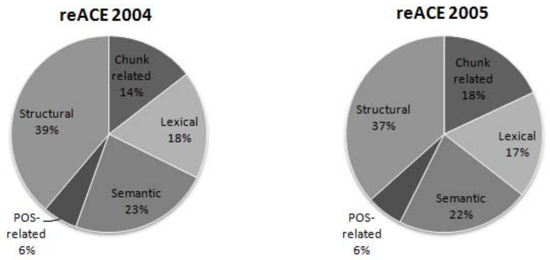

Figure 4. Ratio of the features predicates in the induced set of rules.

E. Error Analysis on reACE Corpora

In all experiments discussed in this section, the typical trade-off between precision and recall was again verified with OntoILPER preferring higher precision than recall. In fact, the ILP learning component in OntoILPER can be biased either towards precision or recall by means of the appropriate parameter setting.

In order to obtain a clear understanding of the precision-recall trade-off in the experiments, the distributions of misclassification errors were evaluated across relation subtypes (Tab. VIII) for both corpora, using all training data. It is worth noticing that, due to the insufficient number of examples, the *business* relation was not included in this analysis.

According to Tab. VIII, the number of *false negatives* (*FN*) overtakes the number of *false positives* (*FP*) for both corpora. A possible reason for that may be due to the unbalanced class distribution in data. In this case, the number of negative examples greatly outnumbers positive examples, with a positive-negative ratio around 10% in both datasets. Thus, for the reACE datasets, in which very few training examples are relation instances, the classifier is less likely to identify candidate instances as actual relations. In fact, the impact of class imbalance on the performance was also reported on the original ACE datasets by [5] and [22]. Reference [3] pointed out that datasets with unbalanced class distributions present a number of problems for all machine learning algorithms. In addition, the domains addressed by RE systems tend to have a large number of relations, in which just a few are positive examples.

This clearly directs the future efforts for a possible solution to the problem of unbalanced class distribution. The most straightforward way of dealing with this problem consists in employing sampling techniques, such as *under-sampling* [3], which selects a subset of negative examples for training. This filtering technique not only allows for the creation of a balanced training dataset by considerably reducing the number of negative examples, but also enables faster rule learning.

TABLE VIII. DISTRIBUTION OF ERRORS IN REACE 2004/2005 DATASETS

	#FP	#FN	Total
reACE 2004	109	145	254
reACE 2005	34	87	121

Another source of errors is related to the introduction of parsing errors and their propagation in the text processing pipeline. As the experimental evaluation over the reACE corpora suggests, shallow natural language processing (tokenization and chunk analysis) is more accurate and may also complement, with useful information, what was missed by deeper text preprocessing techniques, such as dependency parsing.

A final word of discussion concerns the fact that the assessment methodology adopted in the present work does not include comparative evaluations against other RE systems because we could not find any published work using the same corpora we used. However, several comparative experiments of OntoILPER and other state-of-the-art RE system on four RE datasets are reported in [23][24].

VI. Conclusion and Future Work

This paper presented OntoILPER, an ILP-based method for extraction relations instances from textual data. OntoILPER has the advantage of exploiting a domain ontology in its extraction process which opens new opportunities in RE. OntoILPER integrates both a hypothesis space that enables the searching for candidate hypothesis space, and an ILP-based learning component that induces Horn-like extraction rules from annotated examples. We revised the literature and found that many similar ILP-based RE systems do not integrate semantic resources or ontologies in their IE process. Contrastingly, OntoILPER allows for integrating useful background knowledge in many ways, notably by exploiting a domain ontology given as input. OntoILPER was assessed on two standard datasets for RE. The yielded results demonstrated the OntoILPER effectiveness, but there is still room for improvement.

OntoILPER currently relies on shallow syntactic parsing of sentences, which does not take into account semantic aspects relating entities to verbs. Accordingly, we plan to integrate further background knowledge into the preprocessing stage, such as synonyms, hypernym/hyponyms, and semantic role labeling. Our long-term goal here is to fully exploit different types of BK aiming at investigating which kind of BK is more useful on specific domains. Another future line of study concerns the adaptation of OntoILPER on event extraction, i.e., the subtask in IE which tackles the problem of identifying n-ary relations.

Acknowledgment

The authors would like to thank the National Council for Scientific and Technological Development (CNPq/Brazil) for financial support (Grant No. 140791/2010-8) and Hilario Oliveira for his help in the OntoILPER implementation.

References

[1] R. A. Baeza-Yates, and B. Ribeiro-Neto, Modern Information Retrieval: Addison-Wesley Longman Publishing, Boston, USA, 1999.

[2] E. Charniak and M. Johnson. “Coarse-to-fine n-best Parsing and Maxent Discriminative Reranking,” In Proc. of the 43rd Annual Meeting on Association for Comp. Ling. ACL 2005, pp. 173–180.

[3] N. V. Chawla. “Data mining for imbalanced datasets: An overview,” In Data mining and knowledge discovery handbook, Springer, US, 2005, pp. 853-867.

[4] S. P. Choi, S. Lee., H. Jung, and S. Song. “An Intensive Case Study on Kernel-based Relation Extraction,” Proc. of Multimedia Tools and Applications, Springer, US, 2013, pp. 1 -27.

[5] A. Culotta and J. Sorensen. “Dependency Tree Kernels for Relation Extraction,” ACL-2004, July 2004, Barcelona, Spain, 2004.

[6] P. Frasconi et. al. “kLog: A Language for Logical and Relational Learning with Kernels”, Artificial Intelligence 217:117–143, 2014.

[7] B. Hachey, C. Grover, and R. Tobin. “Datasets for Generic Relation Extraction,” Journal of Natural Language Engineering, Cambridge University Press, 2011.

[8] T. Horváth, G. Paass, F. Reichartz, and S.Wrobel.“A Logic-based Approach to Relation Extraction from Texts”, ILP 2009, 2009, 34-48.

[9] P. Kordjamshidi et al.. “Relational Learning for Spatial Relation Extraction from Natural Language,” In Muggleton, S. (Ed.), Tamaddoni-Nezhad, A. (Ed.), Lisi, F. (Ed.), ILP: Vol. 7207, Windsor, Berlin Heidelberg, Springer. 2012, pp. 204-220.

[10] J. Jiang, and C. X. Zhai, “A systematic exploration of the feature space for relation extraction,” Proc. of the Annual Conference of the North American Chapter of the Association for Comp. Linguistics (NAACL-HLT’2007), Rochester, NY, USA, 2007, pp. 113–120.

[11] N. Lavrac and S. Dzeroski, “Inductive Logic Programming Techniques and Application,” Ellis Horwood, New York, 1994.

[12] M-C de Marneffe and C. D. Manning, “Stanford Dedendencies Manual,” 2008.

[13] S. Muggleton, “Inductive Logic Programming,” New Generation Computing 8 (4): 29, 1991.

[14] A. Airola, S. Pyysalo, J. Björne, T. Pahikkala, F. Ginter. All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. BMC Bioinformatics 9: S2, 2008.

[15] G. Plotkin. “A Note on Inductive Generalization”. *Machine Intelligence* 5 , 1971, pp. 153-163.

[16] J. Santos, “Efficient Learning and Evaluation of Complex Concepts in Inductive Logic Programming,” Ph.D. Thesis, Imperial College, 2010.

[17] M. D. S. Seneviratne and D. N. Ranasinghe, “Inductive Logic Programming in an Agent System for Ontological Relation Extraction,” International Journal of Machine Learning and Computing vol. 1, no. 4, 2011, pp. 344-352.

[18] D. Smole, M. Ceh, and T. Podobnikar. “Evaluation of Inductive Logic Programming for IE from Natural Language Texts to Support Spatial Data Recommendation Services,” International Journal of Geographical Information Science, 25, 2011, pp. 1809-1827.

[19] T. Wang et al. “Automatic Extraction of Hierarchical Relations from Text,” In Proc. of the 3rd European conference on The Semantic Web: research and applications (ESWC'06), York Sure and John Domingue (Eds.). Springer-Verlag, Berlin, 2006, pp. 215-229.

[20] D. C. Wimalasuriya and D. Dou, “Ontology-Based Information Extraction: An Introduction and a Survey of Current Approaches,” J. of Information Science, vol. 36, no. 3, June, 2010, pp. 306-323.

[21] G. Zhou, J. Su, J. Zhang, and M. Zhang. “Exploring various Knowledge in Relation Extraction,“ ACL’2005, 25-30 June, Ann Arbor, Michigan, USA, 2005, pp. 427-434.

[22] G. Zhou, M. Zhang, D-H. Ji, and Q. Zhu. “Tree Kernel-based Relation Extraction with Context-Sensitive Structured Parse Tree Information,” Proc. of the 2007 Joint Conference on Empirical Methods in NLP and Computational Natural Language Learning, Prague , 2007, pp. 728-736.

[23] R. Lima, B. Espinasse, H. Oliveira, L. Pentagrossa, F. Freitas. Information Extraction from the Web: An Ontology–Based Method using Inductive Logic Programming. 25th IEEE International Conf. on Tools with AI, ICTAI 2013, Washington DC, USA, 2013.

[24] R. Lima, J. Batista, R. Ferreira, F. Freitas, R. Lins, S. Simske, M. Riss. Transforming Graph-based Sentence Representation toAlleviate Overfitting in Relation Extraction. In Proc. of the 14th ACM DocEng 2014, Denver, Colorado, USA, 2014.

[25] ACE. Automatic Content Extraction. http://www.itl.nist.gov/iad/mig/tests/ace/

[26] ACE 2004. http://www.itl.nist.gov/iad/mig/tests/ace/2004/doc/ace04-evalplan-v7.pdf

[27] ACE 2005. http://www.itl.nist.gov/iad/mig/tests/ace/2005/doc/ace05-evalplan.v2a.pdf

[28] GATE- General Architecture for Text Engineering. https://gate.ac.uk/

[29] LHT Semantic Role Labeler. http://barbar.cs.lth.se:8081

[30] Stanford Core NLP Tools. http://nlp.stanford.edu/software/corenlp.shtml

[31] OpenNLP. http://opennlp.apache.org

[32] The OWL API. http://owlapi.sourceforge.net