Ontotriple: A semantic-oriented symbolic learning algorithm for extracting relation triples

Sanghee Kim, Paul Lewis, and Kirk Martinez

Intelligence, Agents, MultiMedia Group, Department of Electronics and Computer Science,
University of Southampton, U.K.
{sk,phl,km}@ecs.soton.ac.uk

Abstract. A relation extraction system recognises pre-defined relation types between two identified entities from natural language documents. It is important for a task of automatically locating missing instances in knowledge base where the instance is represented as a triple ('entity – relation – entity'). A relation entry specifies a set of rules associated with the syntactic and semantic conditions under which appropriate relations would be extracted. Manually creating such rules requires knowledge from information experts and moreover, it is a time-consuming and error-prone task when the input sentences have little consistency in terms of structures and vocabularies. In this paper, we present an approach for applying a symbolic learning algorithm to sentences in order to automatically induce the extraction rules which then successfully classify a new sentence. The proposed approach takes into account semantic attributes (e.g., semantically close words) as well as linguistic features(entity types) in generalising common patterns among the sentences which enable the system to cope better with syntactically different but semantically similar sentences. Not only does this increase the number of relations extracted, but it also improves the accuracy in extracting relations by adding features which might not be discovered only with syntactic analysis. Experimental results show that this approach is effective on the sentences of the Web documents obtaining 17% higher precision and 34% higher recall values.

Keywords: relation extraction, information extraction, inductive logic programming

1 Introduction

When organisations (e.g. 'museum' or 'gallery') hold an immense quantity of information in the formats of electronic documents or databases, missing values for some data can occur. Examples are the names of people who participated in the creation of an art work or historical events which influenced the artists. To extract such missing values, we might need to rely on additional information sources, like the Web. The Web exists as the largest information repository and new data are continuously added. The observation that most of the Web documents are free-texts in various structures and vocabularies emphasizes the importance of techniques that can extract a piece of information of interest.

Information extraction (IE) systems aims to provide easy access to natural language documents by organising data into pre-defined named-entity types and

relations. Entities can be the name of ‘person’ or ‘organisation’ and ‘location_of’ is an example relation that defines geographical information between two entities. IE systems can rely on extraction rules created from example documents by inducing regular patterns among the examples based on machine learning and/or natural language techniques [4,17]. The accuracy of entity recognition depends on the nature of entity type, for example, ‘painting’ is more difficult to learn than ‘person’ since a ‘painting’ has less distinctive attributes differentiating it from other types. When it is hard to discover consistent patterns among the documents (e.g. Web pages), using gazetteers (e.g. databases of people names) for a pattern-based matching can be an alternative [6]. With respect to end-users efforts, it does not require any extra annotations unless new types of entities are to be learned. One of the shortcomings of this approach is that most entities are pre-defined with specific types such that new entities may not be easily identified.

The task of relation extraction is to extract pre-defined relation types between two identified entities. Many IE systems mainly focus on recognising named-entities (e.g. GATE [6]) and recent experimental results showed that the performance could reach over 80% F-measure [10]. Whereas some systems try to extract relations, the number of relation types is rather small or no relations are extracted [2]. For example, whereas GATE can recognize “*Museum France*” as a type of “*organization*”, but it does not extract the fact that the “*Museum France*” holds a masterpiece of “*Courbet*”. Rather than treating the tasks of entity recognition and relation extraction separately, we regard the relation learning as depending on the entity identifications which provide the conditions under which relations can be extracted. Approaches like [4,18] learn rules for relation extraction over examples annotated by end-users. When the number of relations exceeds dozens, it is infeasible to ask end-users to provide such examples since the users need to annotate a large number of documents.

The relation extraction system tends to achieve a reasonable performance when tested with semi-structured texts or when relations to be learned have distinctive features [18]. For example, ‘date_of_birth’ that holds the date of when a ‘Person’ was born, is easier than that of ‘was_used_for’ which specifies reasons why the art object is important, or its influence on other objects. One of reasons is that there are more ways of describing the latter relation such that it is hard to discover common patterns among examples collected. It is also feasible that more efforts are required to gather examples for such relations. Applying machine learning techniques to derive the distinctive features automatically hence supports a reduction in human efforts to provide such services, or examples. Inducing rules by mining common attributes of the given examples can be supported by using inductive logic programming (ILP), which is a supervised learning system [12]. ILP enables the learner to make use of background knowledge provided and allows the input data to be represented as a prolog style. Since it is based on first-order logic, more complex data structures can be learned. Using the ILP to learn the relation extraction rules hence is suitable for our task.

Semantic variations among the Web documents are considerable since authors use their own styles and vocabularies defining similar statements. Two sentences might be

semantically interchangeable even though they share no similar syntactic tags. Depending on the relation type, it is preferred to extract multiple instances such that the relation extraction rules need to deal with the semantic variations. Extending the rules with semantically close terms can be of use.

In this paper, we describe Ontotriple, a semantic-oriented machine learning algorithm that creates rules for relation extractions. The types of relations considered here are the ones defined in across different syntactic tags (e.g. 'noun', 'verb') such that linguistic analysis as well as semantic understanding is required in extracting such relations. Since the relations extracted are not limited in the descriptions of attributes, the number of relations extracted will be increased. As a supervised learning, Ontotriple needs a set of examples to be marked-up according to the algorithm used. In an effort to reduce workloads on locating and manually annotating the dataset, we use the Web for downloading the examples, and apply a natural language processing technique to automatically annotate them with the algorithm specifications. To cope with semantic variations among the documents, WordNet [11] is used for comparing similarity between two sentences. We evaluate Ontotriple with a small text dataset, and the experiment shows considerably improved performance compared to a simple bag-of-word approach which converts a document into a list of words.

This paper is organized as follows: in section 2, reviews of the related work are given; section 3 describes Ontotriple beginning with an introduction to ILP on which Ontotriple is based, and discusses how to encode texts appropriate for mining. An experimental result is reported in section 4 followed by conclusions and directions for future work in section 5.

2 Related Work

Roth presented a probabilistic method for recognising both entities and relations together [15]. The method measures the inter-dependency between entities and relations and uses them to restrain the conditions under which entities are extractable given relations and vice versa. Local classifiers for separately identifying entities and relations are first calculated. Global inferences are derived from the local classifiers by taking the outputs in conditional probabilities as inputs for determining the most appropriate types. An evaluation with test documents showed over 80% accuracy on entities and a minimum 60% on relations. However, the computational resources for generating such probabilities are generally intractable.

The use of ILP to learn the extraction rules in texts has been attempted in [1, 9, 13]. [9] developed a system that classified e-mail messages into either interesting or non-interesting ones after learning user preferences from e-mail messages read. Message contents were converted into attribute-value pairs describing under which conditions the users are interested in reading a new message. ILP was appropriate for this task since it discovered inter-relatedness among the attributes which were often difficult to

induce with statistical methods, for example, a naïve Bayesian probability. [1] applied ILP to learn relation extraction rules where associated entities are symbols (e.g., 'high', 'low'). It is more concerned with discovering hidden descriptions of entity attributes than creating binary relations between two entities which we are interested in. For example, in the sentence "*Higher levels of CO2 can clearly make plants grow better*", the fact that CO2 has 'high' level can be understood by deducing certain hidden descriptions whereas the Ontotriple has interests in identifying causal relations between CO2 and plants in the sentence. [13] used Progol to learn user preferences concerning WWW pages. Users were requested to rate the pages as either interesting or uninteresting when they browsed them, and then Progol generated a set of rule sets for describing under which conditions users make decisions concerning these classifications. Experimental results showed that Progol achieved a higher or comparable performance to human defined rules.

REES, developed by [3] is a lexicon-driven relation extraction system aiming at identifying a large number of event-related relations. Similarly to the approach here, it depends on a verb for locating an event-denoting clue and uses a pre-defined template which specifies the syntactic and semantic restrictions on the verb's arguments. Ontotriple aims at generating the template automatically from the collected examples instead of relying on knowledge experts or end-users.

Craven et al. implemented the WEBKB project, which aimed to build a knowledge base of Web pages by identifying hidden relationships, which may exist in the pages represented by words and hyperlink definitions [5]. An example of the relations is 'instructors_of(A,B)' which discovers a relationship between course page (A) and instructor's homepage (B) in terms of hyperlink definition. Its main task was to classify Web pages into pre-defined six categories according to the rules created by a rule learning algorithm. To resolve the conflicting predictions that resulted from multi-category problems, a confidence value for each generated rule was computed and compared to other predictions in order to decide which class was assigned.

Ontotriple uses an existing named-entity recogniser (GATE) as well as a lexical database (WordNet [11]) for annotating an entity with pre-defined types. Similarly to the relation extraction, applying machine learning algorithms to induce entity recognition rules has been proposed. [7] uses SRV, a token-basis general-specific rule learning algorithm for information extraction from online texts. It makes use of grammatical inferences for generating pattern-based extraction rules appropriate for HTML structures. Its core token features are separated from domain-specific attributes making the SRV easy to apply to a new system. The evaluation shows lower performance of the multiple-value (e.g. project members) instantiations compared to that of single-value (e.g. project title) entities implying that the former is harder to extract. $(LP)^2$ is a supervised wrapper induction system that generalizes extraction rules based on a bottom-up approach [4]. The generalization starts with word string features suitable for highly structured texts and gradually adds linguistic attributes to induce more appropriate patterns. It uses shallow-level natural language processing, such as POS tagging, or case information ('lowercase'). The generated rules are corrected from mislabeled tags by inducing correct tag positions from a corpus

provided. This correction step is one of contributions that enables $(LP)^2$ to show a higher performance compared to other existing entity rule induction systems (.e.g SRV).

3 Ontotriple

In this section, we present an overview of Ontotriple, a semantic-oriented rule learning algorithm for relation extraction in natural language documents.

3.1 Relation extraction

Ontotriple extracts pre-defined binary relations between two identified entities in a natural language document. The relation is represented as a triple, i.e. predicate(e_1, e_2), where e_1, e_2 are entities. Associated entities restrict the types of arguments to be linked with the predicate. The relation extraction is dependent on the availability of named entities in that mislabelled entities can decrease the number of relations correctly identified. A relation can be implicitly implied in a phrase, for example, [2] extracts an 'employee_of' relation from the phrase of '*an analyst at ING Barrings*', where the analyst is a person type and 'ING Barrings' is an organisation. In this paper, we are interested in relations defined in a sentence-level. An example is '*John works for ING Barrings*', where the verb 'work' links two entities ('John' and 'ING Barrings') with 'work_for' relation. As such, it is necessary to analyse the sentence with natural language techniques both from syntactic and semantic perspectives.

A verb as the central organizer of a sentence posits a core element in recognizing relations between entities. It asserts something about the subject of a sentence for asserting additional information or expresses actions, events, or states of being. For example, in the sentence '*John died on 6th Jan 1900*', the verb 'died' describes an existential status of a person 'John'. As an object, '6th Jan 1900' modifies the verb giving an additional fact of the death event. As such, the verb 'died' acts as a linking word between two identified entities ('John'-person, '6th Jan 1900'-date) and conveys a writer's intention of making the statement. Ontotriple relates the verb to pre-defined relations by considering conditions defined in the relation entry. According to WordNet definitions, each relation is described with a corresponding verb and a sense entry. A word can have multiple meanings (i.e. senses) and it is important to know in which sense the word is used in a given sentence when the semantically close words are collected. Including similar verbs has a purpose of reducing semantic variations between the defined verb and a verb in a given sentence, so that it can increase the number of relations extracted. We rely on Resnik' approach that defines the similarity between two concepts based on the information content of their least common subsumer in a corpus (e.g. WordNet) [14].

3.2. Progol: Inductive Logic Programming

Inducing rules from given examples can be supported by the inductive logic programming technique. ILP can be defined as the intersection of machine learning and logic programming. Contrasted with other learning methods, such as Decision Trees, by using computational logic as the representational mechanisms, ILP can learn more complex, structured, or recursive descriptions and generate the outputs in first-order logic. Learning in ILP is defined with respect to task T, example E (formed as positive or negative) and background knowledge B. A system is said to learn from E and B by constructing a set of hypotheses that can explain new examples. The positive examples are the facts that are true for task T while the negative examples are the facts that should be excluded from the set of hypotheses.

Progol is one of the ILP systems and selects one positive example, constructs the most specific clause and this becomes a search space for the hypotheses [12]. A compression measure is used to compare hypotheses, and is computed by counting the number of positive examples explained, the number of negative examples incorrectly explained, and the number of further atoms to complete the clause. If the hypothesis is confirmed, then Progol looks at the remaining positive examples, and deletes redundant ones. Progol continues until there are no more positive examples left. Input features in the background knowledge are associated with one of three 'mode-type' declarations. '+A' implies that the literal A is an input type in the hypothesis created, '-A' specifying that A is an output variable, and '#A' defines that the literal A is a constant type. This mode type is of use to connect two clauses by allowing one clause to take the output of the other clause as an input.

Table 1: Clauses used for relation learning by Progol

Annotation	Progol clause
Common attributes	has_word(+sentence,-word). has_words(+sentence,-words). consist_of(+words,-word).
Semantic feature	has_verb(+sentence,#verb) has_verbtense(+sentence,#verb,#tense) has_verbmood(+sentence,#verb,#mood) has_subject_word(+sentense,#word) has_subject_words(+subject,#words) has_object_words(+sentence,#words) has_object_word(+sentence,#word)
Named entity	has_entitytype(+sentence,+words,#type) has_gender(+sentence,+words,#gender)
Postag	has_postag(+sentence,+word,#postag). has_postagtype(+sentence,#postype) has_subject_pos(+subject,#postag) has_object_pos(+sentence,#postag)
Word sequence	has_prev(+sentence,+word,#word) has_prevs(+sentence,+words,#words) has_next(+sentence,+word,#word) has_nexts(+sentence,+words,#words)
Word sequence & Postag	has_prev_postag(+sentence,+postag,#postag) has_next_postag(+sentence,+postag,#postag)
Semantic feature & named entity	has_object_type(+sentence,#type) has_subject_type(+sentence,#type)
Semantic feature & Postag	has_subject_pos(+sentence,#postag) has_subject_postype(+sentence,#postype) has_object_pos(+sentence,#postag),has_object_postype(+sentence,#postype)

Since a generalisation is only based on the selected clauses, decisions on how to represent these or what to select normally influence the results. For instance, we have experimented with other representations, and in one of the cases a single clause was tested. It took a long time to construct hypotheses with these complex clauses, and moreover, it required a large number of examples. As such, each relation is represented as simply as possible in a Prolog style. The target clause to be learned is prediction (A,B), where B is the predicted relation entry with which the sentence A is to be associated, e.g. prediction(sentence1, 'place_of_birth'). Each sentence is represented with clauses as described in Table 1.

The common attributes in table 1 are the lists of a single word or words that correspond to the identified entity types. For example, 'Diego Rodriguez' is converted into 'has_words(sen1,'Diego Rodriguez')', 'has_word('Diego')' and 'has_word('Rodriguez')'. It is of use when a concept is referred with different names, as in the case when a person's full name is used first and the first name is cited afterwards. Semantic feature contains clauses about 'verb', 'subject', and 'object' including temporal data of the verb (e.g. 'past', 'present') as well as the way the sentence is to be voiced (e.g. 'active/passive'). Named entities encode the identified entities with available gender information (e.g. 'male/female'). As a result of syntactic analysis, a word is associated with Pos-tagging and in order to reduce the influence of mistagging on relation extraction, 'has_postagtype(+sentence,#postype)' is used for grouping sub-categories of nouns into one type, e.g. 'p-noun' for NNPX, NNPS, NNP. The word sequence defines the ordering of a single word and words. Some annotations are combined in order to test if the combination improves the prediction result. For example, the row denoted as 'Word sequence & postag' in table 1 combines the annotations of 'word sequence' and of 'postag' as well as two clauses specially added to this relation, i.e. 'has_prev_postag(+sentence,+postag,#postag), has_next_postag(+sentence,+postag,#postag).

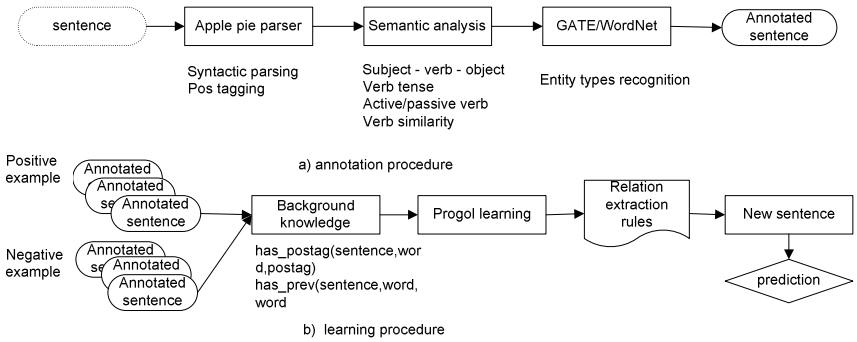

Fig. 1. The procedure for learning relation extractions with Ontotriple

Figure 1 shows the overall procedure of 'Ontotriple', divided into annotation and learning steps. 'Ontotriple' uses the Apple Pie Parser [16] for a syntactic analysis and parts of the semantic analysis tools used in the Artequakt project [8]. Diagram (a)

shows the procedure of annotating a sentence through syntactic, semantic and named-entity analysis. With regard to named-entity recognition, Ontotriple uses GATE and WordNet. GATE provides four different types of entities (i.e. 'person', 'orgranisation', 'date', 'location') and a new type can be recognized if the gazetteers are updated with this information or corresponding extraction rules are created.

A diagram (b) in Fig 1 shows that the background knowledge is constructed from the features of the annotated sentences, and extraction rules are generated by the Progol. The rules then are applied to a new sentence to make a prediction of which relations are most appropriate to be associated with.

4 Experiment

This experiment tests the effectiveness of the proposed approach in identifying relations in natural language documents. It evaluates how automatically generated rules by Ontotriple successfully assign appropriate relations to new sentences. Two used measurements are precision and recall. Precision is the proportion of the correctly predicted relations by Ontotriple to the total number of relations predicted. Recall is the proportion of the number of the correctly predicted relations to the total number of correct relations. A contribution is summarised in the following. Ontotriple is capable of extracting relations from a sentence-level unit without restricting the relations to be defined in phrases. Since it can identify relations in different syntactic tags which might not be feasible with phrase analysis, more various types of relations can be identified. For example, a relation like 'depict_person' is easily extractable in the sentence '*The exterior of the altar depicts Jodocus Vijdt, the donor*' by recognizing the main verb and its meanings. Ontotriple increases the number of the correctly extracted relations by exploring the idea of semantic closeness between a pre-defined verb and a verb in a given sentence. In addition, Ontotriple reduces the effort of experts in manually annotating extraction rules which can be a time-consuming and difficult task. The performance of Ontotriple is compared to that of a baseline which is encoded with a list of words occurring without taking into account either their syntactic roles or semantic meanings.

4.1 Dataset

Supervised learning requires a set of examples to be prepared according to its specifications. The lack of easily accessible datasets restrains the algorithm to be easily applied to new domains. In an effort to reduce workloads due to manual annotations, we used the Web to automatically provide both training and testing documents. That is, one Web site is mined for training, and then we apply the rules generated to a similar site to the one for training. The Web Museum site (http://www.ibiblio.org/wm/about/) has a short biography of some of artists and this site is being maintained well, therefore a dataset was created based on this site. A total of 166 pages were retrieved and downloaded. Each page was analysed following the

annotation steps described in section 3.2. It took over two hours to annotate the sentences. Table 2 shows the relation definitions evaluated in this experiment.

'place_of_birth' is about where a person is born, 'work_in' relates to places where a person did/does work, 'work_as' similarly refers to employee information of a person, and 'represent_person' specifies a subject person that a specific painting is about. Currently, the named-entity recognizer used is not able to extract 'painting' type so that we examined the sentences collected in order to annotate the painting type manually.

Table 2: A relation entry specifying the details of triples used for the experiment

Verb	WordNet sense	Predicate	Entity	Entity
Bear	2	Place_of_birth	Person	Place
Work	4	Work_in	Person	Place
		Work_as	Person	Employment
Represent	9	Represent_person	Painting	Person

As described in section 3.1, to cope with semantic variations, semantic similarity between a target verb in Table 2 and a verb in a given sentence is measured and if it exceeds a threshold (set as 6.0 for this experiment), the sentence is collected. For example, a sentence '*Expressing human misery, the paintings portray blind figures, beggars, alcoholics*' is matched with a predicate, 'represent_person' since the similarity between 'represent (sense '9') and 'portray' is computed as over 7.0.

Progol induces rules over positive and negative examples, and has an option of running only on positive examples (see details in section 3.2). Whereas this option is of use when it is difficult to supply the negative examples, we found that it took a long time to construct the rules with positive examples only and moreover it required a large number of examples. Hence, in this experiment, Progol learns with both positive and negative examples.

A sentence was entered into a training example if the sentence has a main verb corresponding to the verb classes in Table 2. For two aims, we manually examined the sentences collected. First, we removed sentences which are duplicate, or are inappropriate for training. It includes sentences either parsed inaccurately by Apple pie parser or associated with incorrect named-entities. The misclassifications of subject-verb-object identification also cause the sentences to be removed. Secondly, we selected sentences as negative examples when verbs were matched with one of the verbs in Table 2 but have different WordNet senses. For example, a sentence, *'Leonardo was the illegitimate son of a local lawyer (employee) in the small town of Vinci (place) in the Tuscan region'*, is a negative example both for 'Work_in' and 'Work_as' relations. The reason is that whereas 'be' is one of synonyms of 'work' (sense 4), here it is a linking verb that complements the subject ('Leonardo') as described in the object, irrelevant to the 'work' information.

Each annotation described in Table 1 was tried separately with Progol. The following shows examples of generated rules specifying under which conditions a new sentence would be assigned as 'work_as' relation.

```
prediction(A,workas) :- has_word(A,B), has_prev(A,B,a),
has_object_pos(A,'NN').
prediction(A, workas) :- has_subject_postype(A,'noun'), has_object_pos(A,'DT').
prediction(A, workas) :- has_words(A,B), has_entitytype(A,B,'Job'),
has_object_pos(A,'DT').
```

The first example defines that if a sentence has a word pos-tagged as 'NN' (singular noun) in the object and the sentence has 'a' (an article) as a previous word, then the 'work_as' relation is extractable. In the second example, if a subject has a word tagged as 'noun' and if a word tagged as 'DT' occurred in an object, then the sentence is predicted as related to 'work_as'. The third example specifies that if a sentence has an entity tagged as 'job' and if a 'DT' tagged word occurred in an object, then the sentence is related to 'work_as' relation.

4.2. Results

To evaluate the rules generated above, testing documents were downloaded from a Web site called 'A virtual art museum' (http://cgfa.sunsite.dk/) where a list of artist information is retrieval. A total of 88 pages were selected. Negative examples were sorted in the same way as the training examples. A total of 102 ('place_of_birth' (37), 'work_in' (19), 'work_as' (23), 'represent_person' (23)) sentences were used for this evaluation. Table 3 summarizes precision and recall values both of Ontotriple and the baseline. The baseline used only 'has_word(+sentence,#word)' clause, which defines a list of word occurred in the sentence for rule generations. The precision and recall for Ontotriple was the highest value among the predictions made by the five different relation clauses.

Table 3: Precision and recall values comparing the performance of Ontotriple to that of baseline

Relation	Ontotriple		Baseline	
	Precision	recall	Precision	Recall
Place_of_birth	1	0.97	1	0.91
Work_in	0.67	0.75	0.11	0.25
Work_as	1	0.89	1	0.38
Represent_place	0.89	0.82	0.75	0.55
Average	0.89	0.86	0.72	0.52

It is noticeable that the recall value of the baseline was considerably lower than that of Ontotriple except for the 'place_of_birth' relation in which only little difference was observed. On average, Ontotriple obtained 17% higher precision and 34% higher recall. The difference of the precision between Ontotriple and the baseline is most evident for the 'work_in' relation. This relation extracts the description that a person works/worked in a specific location. It is feasible to have erroneous rules with the

baseline since analyzing the sentence only in the perspective of word occurrence is not sufficient to cope with sentences in various formats. Taking into account the following two examples, '*From 1652 Alonso worked mainly in Granada, where he designed the façade of the cathedral (1667)' (*positive sentence*)*, '*During the 1930s he worked in the manner of the Regionalists*' (negative sentence), we can infer that the entity types of the direct object syntax are of use in differentiating the positive example from the negative one. Table 4 shows the detailed performance results of different types of attributes used by Ontotriple.

Table 4: Precision and recall values for the attributes used by Ontotriple

	Annotation	Place_of_birth	Work_in	Work_as	Represent_place
precision	Semantic feature	1	0.33	1	0.47
	Named entity	1	0.57	1	0.78
	Word sequence & postag	1	0.27	1	0.75
	Semantic feature & named entity	1	0.67	1	0.53
	Semantic feature & postag	1	0.25	0.88	0.89
recall	Semantic feature	0.88	0.5	0.28	0.73
	Named entity	0.97	1	0.89	0.64
	Word sequence & postag	1	0.75	0.78	0.55
	Semantic feature & named entity	0.85	1	0.89	0.82
	Semantic feature & postag	1	0.5	0.78	0.73

It is difficult to conclude which annotation performs best across the relations evaluated. For the 'work_in' predicate, using both semantic feature and named entity produces the highest precision and recall values. However, this combination shows lower performance when it is applied to the 'represent_place' relation. This observation is though not surprising since each relation is best characterized with different attributes. For example, the word which pos-tagged with 'IN' (e.g. 'as') is of great use in identifying 'work_as' relation, whereas it is of little use for 'work_in' relation (e.g. 'in') since not only the 'in' word refers to a location, but it also relates to 'style' or 'manner' information. This confirms that it is advantageous to learn rules separately for each relation in order to discover the best strategy.

5 Conclusions and Future Work

We presented an overview of 'Ontotriple' that automatically generates the rules of relation extraction from examples and applies them to classify a new sentence. Ontotriple falls into a supervised learning system and requires a set of examples to be annotated according to the attribute-value pairs defined. A manual annotation is limited in that an expert intervention is needed so that the portability of the approach

can be decreased. Ontotriple uses the Web as a repository of trainable examples and applies natural language techniques to automatically construct the examples. Since there could be errors either in syntactic or semantic understanding including named-entity recognition, human intervention is needed in order to correct the mistaggings. Semantic features as well as syntactic and linguistic descriptions are used for generalising common patterns. The evaluation shows a higher accuracy of Ontotriple compared to the baseline which models a sentence without considering any semantic or syntactic features.

We examined a few issues for further improvement of the proposed approach that could be made in the future. Currently, misclassified pos-taggings or entities are manually corrected in order to use them for training. Similarly to the approach by [4], it might be of use to explore the idea of the 'correct' procedure that re-applies Progol to the generated rules in order to correct mislabeled tags. In Ontotriple, a relation is assigned as a result of classifying a verb in a given sentence into pre-defined verb classes. It is based on the assumption that a verb acts as a core element for conveying the intended statement of a sentence by linking entities. It implies that a relation can be extracted by identifying two entities that are linked by the verb. This is the reason why we collected semantically close sentences by tracing the synonyms of the verb used. However, we observed that there are semantically similar sentences which can not be located by comparing the similarity between the two verbs mentioned. For these sentences, the use of other features, like words in the object tag for measuring similarity might be of use.

6 Acknowledgements

The authors wish to thank the EU for support through the SCULPTEUR project under grant number IST-2001-35372. They are also grateful to their collaborators on the project for many useful discussions, use of data and valuable help and advice. We also thank S. Banerjee and T. Pedensen for software implementing Resnik' similarity measurement approach.

References

[1] Aitken, J. S.: Learning information extraction rules: An inductive logic programming approach, Proc. of European Conference on Artificial Intelligence ECAI, France, (2002), 335-359

[2] Aone, C., Halverson, L., Hampton, T., Ramos-Santacruz, M.: SRA: Description of the IE system used for MUC-7, MUC-7, (1998)

[3] Aone, C. , Ramos-Santacruz, M.: REES: A Large-Scale Relation and Event Extraction System, (2000)

[4] Ciravegna, F.: Adaptive Information Extraction from Text by Rule Induction and Generalisation, Proc. 17th Int. Joint Conf. on Artificial Intelligence, Seattle,(2001)

[5] Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., Slattery, S.: Learning to Extract Symbolic Knowledge from the World Wide Web, In Technical report, Carnegie Mellon University, U.S.A, CMU-CS-98-122,1998

[6] Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V.: GATE: a framework and graphical development environment for robust NLP tools and applications. Proc. of the 40th Anniversary Meeting of the Association for Computational Linguistics, Philadelphia, USA, (2002), 168-175

[7] Freitag, D.: Information Extraction from HTML: Application of a General Machine Learning Approach, Proc. AAAI 98, (1998), 517-523

[8] Kim, S., Alani, H., Hall, W., Lewis, P.H., Millard, D.E., Shadbolt, N.R., Weal, M.W.: Artequakt: Generating Tailored Biographies with Automatically Annotated Fragments from the Web, Proc. of the Workshop on the Semantic Authoring, Annotation & Knowledge Markup conjunction with the Fifteen European Con. on Artificial Intelligence, France, (2002), 1-6

[9] Kim, S., Hall, W., Keane, A.: Natural Language Processing for Expertise Modelling in E-mail Communication, Proc. of the Third Int. Con. On Intelligent Data Engineering and Automated Reasoning, England, (2002), 161-166

[10] Marsh, E., Perzanowski, D.: MUC-7 Evaluation of IE Technology: Overview of Results, available at http://www.itl.nist.gov/iaui/894.02/related_projects/muc/index.html, (1998)

[11] Miller, G.A., Beckwith, R. , Fellbaum, C., Gross, D. ., Miller, K.: Introduction to wordnet: An on-line lexical database. Technical report, University of Princeton, U.S.A.,(1993)

[12] Muggleton, S.: Inverse entailment and Progol, New Generation Computing, 13, (1995), 245-286

[13] Parson, R. and Muggleton, S.: An experiment with browsers that learn, In K. Furukawa, D. Michie and S. Muggleton (Eds.), Machine Intelligence, 15, Oxford University Press, (1998)

[14] Resnik, P.: Using Information Content to Evaluate Semantic Similarity in Taxonomy, Proc. of the 14th Int. Joint Con. On Artificial Intelligence, (1995), 448-453

[15] Roth, D. , Yih, W. T.: Probabilistic reasoning for entity & relation recognition, In COLING'02, (2002).

[16] Sekine, S., Grishman, R.: A corpus-based probabilistic grammar with only two non-terminals, in Proceedings of the First International Workshop on Multimedia annotation, Japan, (2001)

[17] Staab, S., Maedche, A., Handschuh, S.: An annotation framework for the semantic web, Proc. of the First International Workshop on MultiMedia Annotation, Japan, (2001)

[18] Vargas-Vera, M., Motta, E., Domingue, J.: Knowledge extraction by using an ontology-based annotation tool, Proc. of the Workshop on Knowledge Markup and Semantic Annotation, KCAP' 01,Canada, (2001)