

# Axiom-based Feedback Cycle for Relation Extraction in Ontology Learning from Text

Witold Abramowicz\*, Maria Vargas-Vera† and Marek Wisniewski\*

\*Poznan University of Economics, Poland

†Open University, Milton Keynes, UK

**Abstract**—The ontology learning from text cycle consists of the consecutive phases of term, synonym, concept, taxonomy and relation extraction. In this paper, a proposal towards the unsupervised relation extraction method is presented. Our approach is based on text documents and a set of domain axioms which represent the requirements on concepts and relations the user is interested in. We propose to use a feedback cycle that relates linguistic, statistical information and domain axioms. The evaluation is conducted on a corpus describing academic events. However, we believe that the methodology can be applicable in other domains as well. The lessons indicate that the approach is complementary to supervised relation extraction methods and can be used in conjunction with them as a mean to bootstrap an initial ontology.

**Index Terms**—Representation languages, Concept learning, Text analysis

## I. INTRODUCTION

Ontologies are one of the formalism for representing and encoding knowledge. Even though there is no final definition of what an ontology is, there is a widely-accepted definition from the Computer Science community – Gruber’s definition [1]. Many people have interpreted this definition as meaning that an ontology contains explicit static knowledge: some have the view that an ontology is a taxonomy, others that it is a taxonomy plus axioms, and finally a last group believes that an ontology is a set of axioms.

In any case, a static view of ontologies seems predominant and any dynamic aspects are delegated to an application layer that is external to the knowledge representation. The authors believe that an ontology can be defined as an axiomatization of a domain (definition of concepts and relations between them) which acquires new knowledge over time. In other words, changes in the domain conceptualization should be part of the knowledge representation layer. This can be achieved by adding non-monotonicity to the formal system. The reason for this is that, in practice, we cannot expect to always preserve monotonicity in a world where new knowledge can be discovered or facts can change in the domain. This implies that we should be dealing with a non-monotonic artifact. A non-monotonic system allows new axioms to be added (through observations, for instance) that invalidate previously-made inferences. Such non-monotonic reasoning capabilities seem crucial for systems with incomplete information about

their environment. However, for the purpose of this work, we will see our ontology as a monotonic artifact. Provided that monotonicity is assumed in our framework, we have a guarantee that learning a new piece of knowledge will not reduce the set of what is already known.

Our long-term goal is to provide a system which can generate an ontology from a given corpus. Manual engineering of the ontology is a time-consuming task and, in a fast-paced environment, rarely a viable option. Thus, automatic approaches are considered (i.e. ontology learning). Ontology learning from text deals with research problems on a few levels, namely: terms, synonyms, concepts, taxonomic and non-taxonomic relations [2]. In this paper, the problem of axiomatization within concepts and non-taxonomic relations is addressed.

### A. Related Work

A substantial number of proposals for ontology learning from text already exists. These proposals are usually classified in terms of affinity to one or more tasks within ontology learning from text [3]. In our motivation, we only consider the class of approaches that refer to relation extraction for ontology learning from text.

Some of the well-known approaches, namely OntoLearn [4], OntoLT [5] and Text2Onto [6] were created with the purpose of assisting users with the task of ontology creation with a given corpus. However, their main drawback is that the ontology creator is assumed to be a domain expert, i.e. she is assumed to know exactly what kind of relations are relevant for the domain. For this task, both of these tools provide specialized methods, e.g. OntoLT allows user for formalization of linguistic rules based on XPath expressions [7].

Some of the other methods require the user to manually create a sample set of relations in a method-specific form.

This problem has been recently pointed out in the literature. Work presented in [8] tries to tackle this by proposing the kernel-based methods to minimize supervision. However, the main drawback stays the same – the user is still required to be a domain expert, even for stating a few positive and negative examples. In practice, it means that it is extremely difficult to run the extraction process. Other approaches to the problem focus on integration of different sources. For instance, [9] introduce popularity and uniqueness measures for relation classification.

In this paper, we propose an approach that releases the user from providing even the smallest set of examples.

This work has been partially supported by Marie Curie Transfer of Knowledge Fellowship of the European Community’s Sixth Framework Programme under the contract number MTKD-CT-2004-509766 (enIRaF).

## B. Contribution

Our main contribution is the proposal of a method for the recognition of relations from text using an axiomatic approach.

**Thesis.** *The information necessary to extract non-taxonomic relations and instances in ontology learning from text is not only provided by linguistic annotation sets but also within the existing ontology state and domain specific rules, i.e. axioms. Moreover, both linguistic annotation and current ontology state can be inferred from one another. We therefore propose a feedback cycle to feed instances and non-taxonomic relation extraction with linguistic information and current ontology state by using domain axioms.*

The lack of approaches to this problem means that a comparative evaluation is not possible. In such cases, researchers often modify slightly the evaluation baseline so that other approaches fall into the same category. Instead, we opt for a more methodological evaluation with a qualitative approach that gives a clear advantage over existing methods and still yields a reasonable quantitative results. As a result, we do not provide detailed insights into other relation extraction methods, such as frequency-based or lexico-syntactic approaches, nor do we examine the pros and cons of these methods.

This paper is organized according to IMRAD style: in the next section the method for axiom-based feedback cycle is presented. In Sect. III a test bed and evaluation results are provided. Finally, Sect. IV highlights the main conclusions.

## II. METHOD

Let us define the meta-model of ontology learning from text  $M$  as a tuple:

$$M = \{D, T, C, TR, A, NTR\} \quad (1)$$

$D$  is a set of documents  $D = \{d_1, \dots, d_n\}$ . The set  $D$  forms a corpus for the whole process of ontology learning.  $T$  is a set of terms  $T = \{t_1, \dots, t_n\}$  conforming to the definition in ISO 1087-1:2000 that is extracted from  $D$  with term extraction functions, mostly by using clearly-defined linguistic annotation sets.  $C$  is a set of concepts  $C = \{c_1, c_2, \dots, c_n\}$  that is extracted from the set of  $T$ , mostly by using linguistic resources.

$TR$  is a set of taxonomic relations that hold between elements of set  $C$ :  $TR = \{isa(c_{i1}, c_{j1}), \dots, isa(c_{in}, c_{jm})\}$  where  $c_{i1} \dots c_{in}$  is the set of subjects (heads) of  $isa$  relation,  $c_{j1} \dots c_{jm}$  is the set of objects of  $isa$  relation and  $\forall isa i \neq j$ .

The operations performed on the first four elements of the meta-model result in a hierarchy of concepts. Our method is concerned with, and focuses on, the last two elements.

$A$  is the set of axioms:

$$A = \{A_1 \Rightarrow B_{11}, B_{12}, \dots, B_{1m_1}; \dots, A_n \Rightarrow B_{n1}, \dots, B_{nm_j}\} \quad (2)$$

where each  $A_n$  is a head of a clause and each  $B_{n1}, \dots, B_{nm_j}$  is the body of the clause. The set of  $m_1 \dots m_j$  consists of body elements that for each  $A_n$  can be different.

Finally,  $NTR$  is a set of non-taxonomic named relations:

$$NTR = \{rel_1(c_{x_1}, c_{y_1}), rel_2(c_{x_2}, c_{y_2}), \dots, rel_n(c_{x_n}, c_{y_n})\} \quad (3)$$

where  $rel_1, \dots, rel_n$  is a set of non-taxonomic relations,  $c_{x_1}, \dots, c_{x_n}$  are subjects of  $NTR$  and  $c_{y_1}, \dots, c_{y_n}$  are objects of  $NTR$ . Each of the  $NTR$  is a domain-dependent relation. For example, the semantics of the relation  $worksAt(x, y)$  should be interpreted as a binary relation which takes the first argument from the set of *Person* and the second argument from the set of *Organization*.

### A. Axioms

In our method, we utilize standard domain knowledge in the form of axioms  $A$  (2). Axioms are treated as requirements of the domain expert and tell us what kind of entities to extract. For instance, if domain axioms state that "worksAt" relation is important for the domain application, it is our goal to use any axioms related to this relation and extract related entities.

Therefore, we release the user from being a domain expert. This is particularly important, as this knowledge is often quite specialized and the user may well not know what is relevant.

For analysis purposes, we have conducted an observation test. The goal was to discover the possible axioms for the domain. In our example domain, the domain rules would include:

```
UniversityStaff(x) => worksAt(x,y) AND University(y)

hasTitle(x,y) => Person(x) AND Title(y)
AND co-occurInTerm(x,y)

Professor(x) => Person(x)
AND includesTerm(x, oneOf(Professor,Prof,Prof.))
```

Axioms are represented in a variant of first-order logic that can be reduced to Horn clauses. As Horn clauses are deduced by most reasoning tools, we have decided to use Horn logic to represent the axioms that are used in our approach.

For technical interoperability issues, we are using Semantic Web Rule Language (SWRL) as a serialization language for axioms. Although SWRL has some additional primitives, it is considered to be compatible with Horn logic<sup>1</sup>. This choice is also motivated by the fact that ontology reasoning frameworks support SWRL.

### B. Feedback Cycle

The feedback cycle for ontology learning from text is a cyclic process which includes:

- using linguistic annotation features to create new ontology statements;
- using domain knowledge in the form of axioms to create new ontology statements;
- allowing for linguistic annotation features to be used in axioms;
- allowing for axioms to be used in linguistic annotation features extraction.

The first two features are straightforward – they allow for using standard sources for ontology learning. The other

<sup>1</sup><http://www.w3.org/Submission/WRL-related/>

two allow for using both of these sources alternatively. It is possible to state the linguistic annotation features in axioms and use information from axioms in linguistic annotation features extraction. Therefore, a feedback cycle is created in which knowledge from one of these sources can be used in the other one.

1) *Requirements*: The proposed process needs to significantly improve over existing types of tools. Requirements for the solution come from an analysis of domain axioms which uncovered the need for three additional types of predicates, namely:

- Occurrence of a given token – a lexical realization of an instance includes the label of ontology Class, e.g. includesTerm(x, "Department").
- Co-occurrence of two tokens within single term – two labels of instances occur in single term, e.g. Person(x) AND Title(y) AND co-occurInTerm(x,y).
- Co-occurrence of two terms within single document – two labels of instances co-occur in documents, e.g. Professor(x) AND University(y) AND co-occurInDocument(x,y).

Semantics of all three additional predicates relate to linguistic features of their arguments. Therefore, we shall call them linguistic predicates.

2) *Solution*: Two methods to resolve additional linguistic predicates were considered:

- **In-line annotations**. Place annotations before each rule in the SWRL file and resolve them by using specific processors. For instance, co-occur(x,y) is not inside the swrl:Imp element in SWRL but within a specific tag before the rule, let it be "Processing information". Before using the reasoner, the specific processor processes this information and finds co-occurrences of x and y within documents. After finding some of them, the processor should rewrite the rules by adding predicates of the found types. The main effort in this solution is to write the processor that would be capable of resolving the rules.
- **Object properties**. Treat the additional operations as standard ontology object properties. The main effort is to resolve all these additional predicates before using the reasoner.

The latter method is more natural, as it resolves all the additional elements to standard ontology objects. It is also more transparent, since the rule itself consists of all the constraints and none of them are hidden behind the processor's internal logic. Therefore, we propose the latter approach which consists of the following steps:

- 1) Each proper ontology should include additional object relations:
  - a) includesTerm(owl:Thing, owl:Class).
  - b) coOccurInTerm(owl:Thing, owl:Thing),
  - c) coOccurInDocument(owl:Thing, owl:Thing),
- 2) For a given corpus, before conducting the reasoning, the additional object properties must be resolved to point to proper corpus objects. For instance, in order to resolve occursInTerm linguistic predicate, we must look for all lexical occurrences of a given class within token lists of

terms. The information necessary for these operations is found within linguistic annotation sets specific to each application. As a result, it is known in which instances a given class occurs.

- 3) The ontology model is populated with realizations of linguistic predicates for instances.
- 4) The reasoning is run on the submitted rules list.
- 5) New axioms are added to the non-monotonic ontology model.

The domain and range of arguments owl:Thing in step 1 should be further constrained to the specific types of resources used in the application of the method. Such operation shall greatly limit the size of the objects to be considered by the reasoner, and therefore limit the computation complexity of the method and boost performance.

Each of the linguistic predicates has a different semantics and needs a separate implementation. The implementation effort varies significantly. For demonstration purposes, we have implemented all of the three predicates. Occurrence within a term is the simplest to implement and compute, since it only requires checking all terms within a corpus to be checked once for an occurrence of a specific strings (classes labels). The complexity of co-occurrences predicates is much higher, because it requires the application of standard co-occurrence measures such as Jaccard, Dice or Cosine measures. This is especially true when calculating co-occurrences in documents where the computation complexity is high. Therefore, we strongly suggest pre-calculating co-occurrence measures and applying methods on the already-calculated scores. In our method, all three measures are calculated, but only the Jaccard measure is utilized.

Both ontology model and rules serialization files tend to be large. Mixing ontology model with rules within a single file is error-prone and complex in management. Therefore, the rule file in SWRL, as a proper OWL file, should import the ontology model stored in the separate file. As a result, cohesion is improved because both of these files are used in different phases. The ontology model is created and changed in the ontology learning phases and the rules file is modified when domain axioms change.

### III. RESULTS

To conduct an evaluation we had to provide: corpus, seed ontology with concepts and a set of domain axioms.

#### A. Test Bed

1) *Corpus*: The corpus used as our test bed was the KM<sub>i</sub> (Knowledge Media Institute) electronic newsletter. The KM<sub>i</sub> newsletter is used to share information between members of organizations. The text used in the news articles is unstructured. For example, the writers sometimes use slang, break conventions for capitalization, incorrectly use punctuation, etc. Also, styles of writing in the archive can vary tremendously, as news are submitted by dozens of individuals writing in their personal writing styles. Moreover, some are not native speakers of English and peoples backgrounds range from multimedia to formal languages.

The KMi corpus consists of 273 text documents (62303 tokens) which gives an average of over 228 tokens per document.

2) *Seed Ontology*: Classic ontology learning from text process is sequential. Therefore, in order to evaluate the relation extraction task, we have had to conduct the extraction process of terms, synonyms and concepts. For term extraction, we used Machine Learning algorithm based on n-gram model, for synonyms classification – simple disambiguation method based on WordNet, for concept formation – simple intension test and for Named Entity Recognition – the default ANNIE linguistic resources. For a fair tutorial on these methods, please refer to [3] and relative tool documentation.

As a result, we have bootstrapped an initial ontology (seed ontology) that is used as an input for our core experiments and consists of:

- 417 classes, e.g.: Colleague, PilotProject, University;
- 579 instances of types specified in standard ANNIE’s NE tags;
- labels for instances that correspond to the lexical occurrence of terms in text.

3) *Axioms*: The general rules for academic domain were provided by an expert that did not have access to the KMi corpus but knew the organisation well<sup>23</sup>. Expert indications were placed into a single file and consisted of 25 rules. These belonged to the following axiom types:

- 12 rules include only standard predicates;
- 6 rules include includesTerm linguistic predicate;
- 6 rules include coOccurInDocument linguistic predicate;
- 1 rule includes coOccurInTerm linguistic predicate.

## B. Linguistic Predicates

As explained in Sect. II, the seed ontology was populated with realizations of linguistic predicates. This process added to the seed ontology the following properties:

- 110 includesTerm object properties;
- 28 coOccurInTerm object properties;
- 547-26304 coOccurInDocument object properties depending on a threshold of the Jaccard co-occurrence measure.

The first two object properties were created based on the lexical occurrence of arguments. The CoOccurInDocument object property was calculated based on the Jaccard co-occurrence measure. The relation among all instances was analyzed and co-occurrence scores were pre-calculated. The exact number of properties instances depending on a threshold is depicted in Tab. I.

According to our recommendations, the domain and range of all linguistic predicates have been constrained to the classes actually used.

Unfortunately, experiments showed that the Pellet reasoner cannot handle our ontologies. Despite the authors’ performance tests and optimizations [10], we proved in our test

<sup>2</sup>We had the pleasure to welcome a KMi researcher from Open University as a guest at our university.

<sup>3</sup>In most cases, the rules in Horn logic were already present or were easily generated or transformed from other representations.

0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.8	1.0
26304	5550	2136	1235	976	931	591	551	547

TABLE I  
DISTRIBUTION OF THE NUMBER OF COOCCURRENCEINDOCUMENT LINGUISTIC PREDICATE INSTANCES.

bed that using in one run Jena + Pellet + SWRL rules + ontology imports and ontology consisting of 417 classes, 579 instances and up to 26304 object relations among instances is not possible. Only by reducing the number of instances and object properties, were we able to conduct our tests.

As a first pruning operation, we divided the tests into the set of subtests devoted to linguistic predicates. As a result, the subtests were performed on three linguistic predicates, namely includesTerm, coOccurInTerm and coOccurInDocument.

In the includesTerm and coOccurInTerm linguistic predicates subtests, we pruned from the test ontology all instances without the linguistic predicate. This pruning operation does not influence the results of subtests but reduces the total number of instances in includesTerm subtest to 82 and in coOccurInTerm subtest to 28. The difference between 110 (number of includesTerm object properties) and 82 is based on the fact that some of the instances co-occur with more than one term.

In the coOccurInDocument linguistic predicate subtest, we pruned the test ontology on the level of each tested rule. This was motivated by the fact that pruning the ontology only by pruning instances without the linguistic predicate does not yet result in an ontology that can be handled by Pellet. Additionally, we had to prune other instances that do not directly participate in a given rule reasoning. Therefore, a possible subset of instances to prune was specific to a rule itself. For instance, the collaborate rule does not need any other instances than Organizations and Universities: this suggests pruning of all Person instances which halves the number of ontology instances.

Rules with standard predicates were not evaluated because of two performance reasons. Firstly, the predicates of these rules relate to much of the total number of instances. This means that, to conduct a sound evaluation, it would be needed for Pellet to handle approximately 500 instances and we know this not to be possible. Secondly, these rules were based mainly on theorems that were the results of axioms consisting of linguistic predicates. The evaluation of standard predicates requires other axioms with linguistic predicates to be resolved which would also raise the complexity of the model and further strengthen the performance problems.

## C. Evaluation Measures

The direct results of our approach are new axioms that are deducted from the set  $A$  (Eq. 2). These results are compared to the indications of an expert (i.e. golden standard or baseline). The golden standard was retrieved from an expert group during series of workshops. The resulting baseline consists of manual indications of what relations are important and the list of their instances. For instance, the expert group indicated that the relation worksAt(x,y) is relevant and, based on the text,

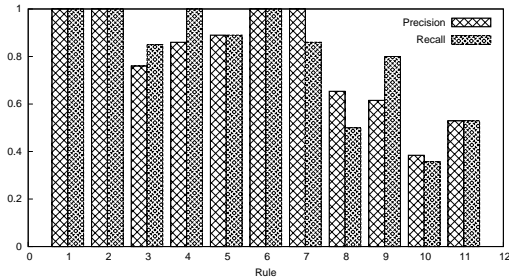


Fig. 1. Precision and recall of the approach with a distribution into rules. Rules 1-7 include includesTerm and coOccurInTerm linguistic predicates, rules 8-11 include coOccurInDocument linguistic predicate with an optimized threshold.

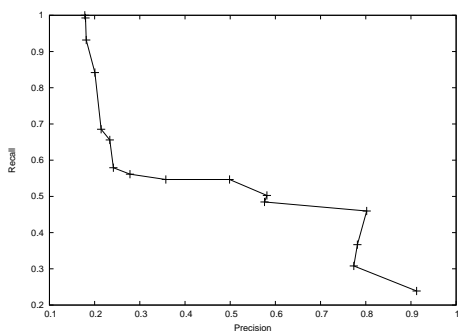


Fig. 2. Rules with coOccurInDocument linguistic predicate (8-11) based on 26 probability thresholds (0; 0,01, 0,02, ... 0,1; 0,12; ... 0,2; 0,25; 0,3; 0,33; 0,4; 0,5; ... 1).

indicated each pair of concepts from the seed ontology that meets this relation. The indications of the expert group and method results were calculated to produce precision and recall measures.

The results are presented on two figures. Figure 1 depicts the precision and recall of all rules. Rules with includesTerm and coOccurInTerm linguistic predicates (1-7) are calculated based on binary search, therefore the presented scores are not dependant on the probability threshold. The other scores refer to rules with coOccurInDocument linguistic predicate with the threshold manually set to a level with the best trade-off between precision and recall. The difference between 25 rules as indicated in Sect. III-A3 and 11 presented is caused by a deduction of rules with standard predicates and additional 2 rules that did not occur in the text.

Figure 2 depicts the distribution of average scores for rules with the coOccurInDocument linguistic predicate which are based on 26 probability thresholds ranging from 0 to 1.

#### IV. DISCUSSION

The main contribution of the presented method is to release the user from a need to supervise the process of relation extraction in ontology learning from text. The user is not forced to produce any linguistic, domain-specific rules nor even the smallest set of positive and negative examples. The knowledge of the domain is extracted from the domain axioms

that are domain-general. We argue that this is a substantial step towards unsupervised relation extraction methods and outperforms former approaches to the problem presented in Sect. I-A.

The quantitative evaluation depicted on Fig. 1 and 2 was performed to produce reasonable levels of precision and recall. We have learned that it is feasible to bootstrap an ontology mainly from domain axioms. We also think that the proposed approach is complementary to other approaches that require a user supervision and can be successfully utilized as a starting point for these approach that, in general, produce a high precision level but cause problems with a recall.

The decision whether our method would outperform other approaches is application specific. If the user is comfortable with examining a text and producing an example set of linguistic rules, we do not pretend that our method will be more suitable for her. But if the user cannot afford to spend time on linguistic specifics, our method will be better. We strongly believe that the latter situation happens more often than the former.

We have greatly suffered from the limitations of current reasoners – further advances in this domain (automatic reasoning) would produce better results; results from the first iteration could be submitted to the next, thus realization of a feedback cycle could be fulfilled. With current reasoners advances, we are only able to produce single iterations. Pellet was used mainly because its Jena integration. We did not experiment with other reasoning systems, but we believe that some of the Pellet’s limitations could be avoided by using more efficient reasoners, such as FACT++ or RacerPro.

As future work, we plan to investigate the possibility of exploiting more complex NLP tasks, such as concept disambiguation and relating them to concept instances within axioms.

#### REFERENCES

- [1] T. R. Gruber, “A translation approach to portable ontology specifications,” *Knowl. Acquis.*, vol. 5, no. 2, pp. 199–220, 1993.
- [2] P. Buitelaar, P. Cimiano, and B. Magnini, “Ontology learning from text: An overview,” in *Ontology Learning from Text : Methods, Evaluation and Applications*, P. Buitelaar, P. Cimiano, and B. Magnini, Eds., 2005.
- [3] P. Buitelaar and P. Cimiano, “Ontology learning from text: Tutorial,” in *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, 2006.
- [4] M. Missikoff, R. Navigli, and P. Velardi, “Integrated approach to web ontology learning and engineering,” p. 3, 2002.
- [5] P. Buitelaar, D. Olejnik, and M. Sintek, “A protege plug-in for ontology extraction from text based on linguistic analysis,” in *Proceedings of the 1st European Semantic Web Symposium (ESWS)*, 2004.
- [6] P. Cimiano and J. Vlker, “Textonto - a framework for ontology learning and data-driven change discovery,” in *10th International Conference on Applications of Natural Language to Information Systems*, 2005.
- [7] M. Sintek, P. Buitelaar, and D. Olejnik, “A formalization of ontology learning from text,” in *International Semantic Web Conference*, Hiroshima, Japan, 2004.
- [8] R. Bunescu and R. Mooney, “Learning to extract relations from the web using minimal supervision,” in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Association for Computational Linguistics, June 2007, pp. 576–583.
- [9] J.-X. Huang, J.-A. Shin, and K.-S. Choi, “Integrating relations for a domain ontology,” in *Proceedings of the 6th International Semantic Web Conference*, Busan, Korea, Nov. 2007.
- [10] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, “Pellet: A practical owl-dl reasoner,” *Journal of Web Semantics*, vol. 5, no. 2, 2007.