Using Ontology Alignment for the TAC RTE Challenge

Reda Siblini and Leila Kosseim
CLaC Laboratory
Department of Computer Science and Software Engineering
Concordia University
1400 de Maisonneuve Blvd. West
Montreal, Quebec, Canada H3G 1M8
r_sibl,kosseim@encs.concordia.ca

Abstract

In this paper we present the system that we used to participate to the TAC RTE challenge. The system relies on acquiring and aligning ontologies to recognize textual entailment. It automatically acquires an ontology representing the text fragment and another one representing the hypothesis, and then aligns the created ontologies. By learning from available textual entailment data, the system can then classify textual entailment using the information collected from its ontology alignment phase. The system performance evaluation achieved an accuracy of around 68% on the two-way task, and an accuracy of 61% on the three-way task when compared to human annotators.

1 Introduction

Textual entailment is defined by the RTE challenge as: "a directional relationship between two text fragments, which we term the Text (T) and the Hypothesis (H)", where "T entails H if the truth of H can be inferred from T within the context induced by T". We attempt to recognize textual entailment by acquiring a formal semantic representation from T, and another one from H. The formal representation is a description logic based ontology. Consequently, we reduce the textual entailment problem to an ontology alignment one, as we believe that by aligning the two acquired ontologies we will be able to discover whether T entails H or not.
An overview of our system is given in Section 2, a more detailed description of our ontology acquisition and alignment methods is provided in Sections 2.1 and 2.2, then we describe our textual entailment method in Section 2.3, and we provide our performance evaluation results on the RTE challenge in Section 3.

2 System Overview

This was our first participation to the RTE challenge, and we participate with a newly built system that allows the recognition of textual entailment through three main phases:

- The first phase is the ontology acquisition phase, which is responsible of acquiring a formal representation of the supplied text's concepts and their relationships. This phase produces two ontologies: an Ontology-T representing the Text's ontology, and Ontology-H representing the Hypothesis's ontology.
- The second phase is the ontology alignment phase. In this phase, the system aligns the ontologies created from the previous phase into one ontology that we refer to as Ontology-A, this phase also output the alignment information that will be used in classifying textual entailment.
- The last phase of the system is the textual entailment phase. This phase relies on the information collected from the ontology alignment phase to decide whether the resulted alignment information conveys an entailment of the hypothesis from the text or not. The decision is made using different machine learning algorithms that have been trained over similar alignment data collected from pairs of text and hypothesis and their textual entailment results.

Figure 1 displays a diagram of the system's architecture, showing in details the main phases of the system, the subparts, and its interaction with other systems. The next section will go in more detail into the ontology acquisition phase.

2.1 Ontology Acquisition

The ontology acquisition phase is an automated process where the system computes a formal semantic representation of the supplied text. This process includes many steps: First a syntactic analysis step that will syntactically parse the provided text through the Minipar parser [Lin, 1998], resulting in a set of dependency relations of the form:

`Grammatical relation (Governing Word, Modifying Word)`

These relations are then processed through a set of transformation rules in order to create a semi-semantic structure of the form:

`Relation (Governing Content Word, Modifying Content Word)`

The difference between the dependency set and the semi-semantic structure is restricting Governing and Modifying words into Content words. This restriction will give more meaning to the relations, and will transform the grammatical relation to a general one, which will include relations inherited from function words.
The second step is semantic analysis, where we transform the semi-semantic structure to what we refer to as an Intervaria Semantic Structure (ISR) of the form:

`Relation (Governing Verb, Content Word)`

The main difference between the two structures is that the governing word will be restricted to the verb class, which will introduce additional meaning that is usually implicit in a sentence.

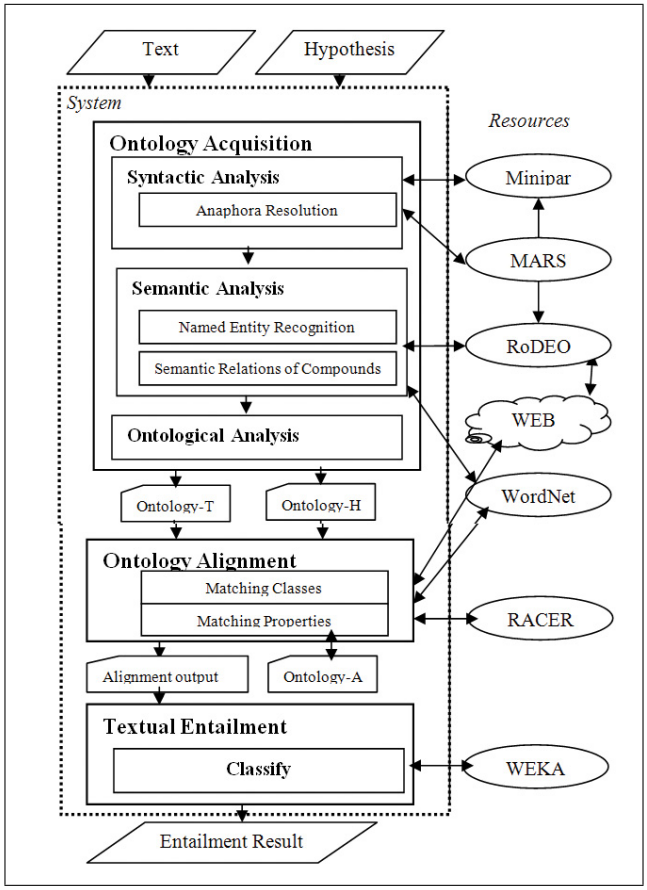

Figure 1: System Architecture

The last step is the ontological analysis. This step will transform the ISR structure to a description-logic based ontological structure. This is done by transforming the Relation into a Property, the Governing verb into the property's Domain, the Content word into the property's Range, and creating related individuals.

2.1.1 Syntactic Analysis

The syntactic analysis phase begins with syntactic parsing of the provided text using the Minipar parser. The parser output consists of a set of dependencies, each having the following information:

```
[Modifying word] (Part of speech) (Stem)
[Grammatical relationship] [Governing word]
```

The parsed output is then processed through a set of transformation rules that have been created to add a restriction on the governing and modifying words to belong to the

content words or lexical category (verb, noun, adjective, or adverb). This restriction will give additional meaning to the structure, and will change non-content words to be either feature of the governing word, or as the actual relationship. We refer to this structure as a Semi-semantic structure. The Semi-semantic structure could be described through the following format:

```
Relationship (Governing Content Word |Feature(F)|
Part of Speech(POS), Modifying Content Word| feature|POS)
```

2.1.2 Semantic Analysis

The main purpose of the semantic analysis is to further restrict the semi-semantic structure created from the previous step by transforming it to an Intervaria Semantic Representation (ISR) having the following format:

```
Semantic Role (Governing Verb - stem - feature - POS,
Content Word - stem - type - feature - POS)
```

The main differences between the two structures are the restriction of the governing word to the verb class, and the addition of the content word type or hypernym.
We are dealing with syntactic variations by restricting the representation into specific syntactic categories. In order to create such a structure, we need to deal with the extraction of named entity classes to be able to find the semantic relation that could exist between noun compounds, and then to discover such semantic relation that could be described by a verb. The next sections will go over these subtasks in more details.

Named Entity Recognition The first step in creating the semantic representation is finding out the types or classes that named entities belongs to. This information is required in order to be able to identify verb characterizing the semantic relation between noun compounds, and to add more meaning to the presentation. To do so we use the method described by [Sibini and Kosseim, 2008]'s to extract type of named entities from the web. However, before doing so we check the availability of such information in the semi-semantic structure, such as the availability of predicates, or appositives that usually describes a named entity. For example in the sentence *"Jurassic Park is a novel"*, the type of the named entity *"Jurassic Park"* is already found in the text to be a *"novel"*. However we still need to get the classes that describe a named entity such as *"Michael Crichton"*. Using [Sibini and Kosseim, 2008]'s system we retrieve the following list of classes, among others, that describe *"Michael Crichton"*: *writer, director, producer, winner...*

Semantic Relations of Compounds In the transformation of semi-semantic structure to a semantic one, we have restricted the governing word to the verb class, hoping to limit syntactic variations and to give more meaning to the structure. So what should we do about nouns or adjectives modifying other nouns? One solution would be to find a verb that could describe the semantic relationship between them. A verb will preserve the structure that we are looking for, and will make explicit the implicit commonsense knowledge available between the compounds. For example, the following semi-semantic sub-structure of a hypothesis example: `of (Michael Crichton|writer,`

`Jurassic Park|book)` requires a verb to describe the relationship between the two named entities. Using the previous task we know that *"Michael Crichton"* is a writer, and that *"Jurassic Park"* is a book. Using the RoDEO system of [Sibini and Kosseim, 2008] and a predefined set of grammatical patterns, we could find the verbs that characterize the semantic relation between *"book and writer"*, which in turn describes the relationship between the two named entities. For this particular example, the system have retrieved the following list of verbs (with their frequencies): *write(1827), have(1583), promote(826), print(432)....*
By selecting the highest ranking verb, we could divide the semi-semantic tuple into the following:

`subject (write, Michael Crichton|noun-writer)`

and

`object(write, Jurassic Park|noun - book)`

In addition to using a verb to describe semantic relations of noun-compounds within a sentence, we also use it to describe semantic relations between sentences by using a simple heuristic to check for the semantic relatedness of one noun and a list of candidate nouns that precede it in a text, and within a predefined lexical distance. We used the WordNet::Similarity [Pedersen et al., 2004] Path Length method as the semantic relatedness of two nouns. The path length method is a simple node-counting scheme, which returns a relatedness score between two concepts. The score is inversely proportional to the number of nodes along the shortest path between the synsets in WordNet.

2.1.3 Ontological Analysis

The main purpose of the ontological analysis is to translate the ISR representation created from the semantic analysis phase to a formal semantic structure. Following from the semantic analysis phase, the ISR structure has the form:

`Relation (Governing Verb [stem - features - POS],`
`Content Word [stem - type - features - POS])`

that will be translated to the ontological form:

`Property (Domain, Range)`

Where the `Governing verb` and `Content word` will be both translated into `OWL:classes` and indexed instances accordingly. The `Relation` will be translated into `owl:property` having the `Governing verb` as its `Domain`, and `Content word` as its `Range`. All classes will be subclasses of their stem, and of their part of speech. In addition, the `Content Word` will also be a subclass of its `Type`. The method is a bottom-up approach, where the governing verb and the content word result in an instance labeled *"Word-index"* that is generalized to a concept labeled *"Word"*, where index is a numeric value, needed to make each instance of the word a unique instance. On the other hand, each property results in a sub-property labeled *"Word-index"* of the property labeled *"Word"*, so that the domain and range are related to the sub-property only, making the generalization on the sub-properties and not the property itself.

To illustrate the above, let us take the following text-hypothesis pair example, which was taken from the RTE3 [Giampiccolo et al., 2007] development set:

(T): Jurassic Park is a novel written by Michael Crichton....
(H): Michael Crichton is the author of the book Jurassic Park.

A graphical illustration of the acquired ontology-T from (T), is shown in Figure 2, and the acquired ontology-H from (H) is shown in Figure 3. In these graphs beveled-edged rectangles represent OWL classes, perpendicular-edged rectangles represent OWL properties, polygons represent a sub-properties or a sub-classes depending on what they are connected to. Note that instances are not shown in these graphs. [1]

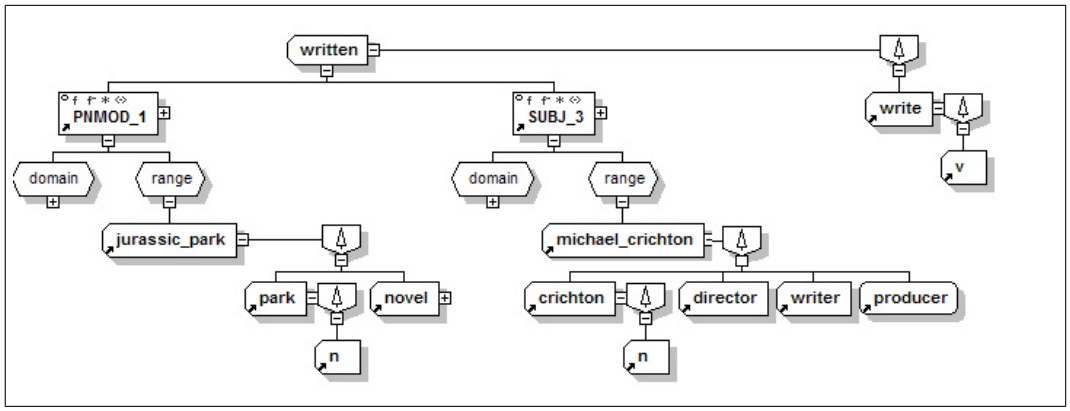

Figure 2: Acquired Ontology-T

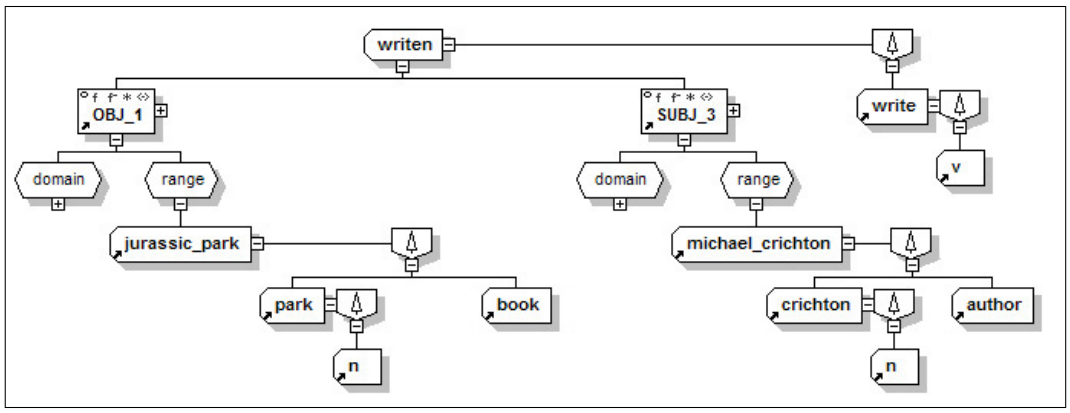

Figure 3: Acquired Ontology-H

[1]The ontology graphs were automatically generated from OWL code using SemanticWorks, `http://www.altova.com/products_semanticworks.html`

2.2 Ontology Alignment

The ontology acquisition phase will result in two ontologies: Ontology-T representing the supplied Text and Ontology-H representing the supplied Hypothesis. The ontology alignment phase is responsible of aligning the created ontologies into one ontology that we refer to as Ontology-A. The alignment could be seen as on-demand tuning of the created representation in order to acquire missing information and align available ones. The alignment is done in two steps, the first step matches the classes of Ontology-H to the classes of Ontology-T, and creates equivalent classes when necessary. The second step matches the properties of Ontology-H to the properties of Ontology-T, and creates equivalent properties when necessary, resulting in an Ontology-A.

Figure 4 illustrates graphically the resulted aligned ontology: Ontology-A. Note that the hexagon having an equal sign and a rectangle describes an equivalent property, and the hexagon having an equal sign and a beveled-edged rectangle describes an equivalent class.

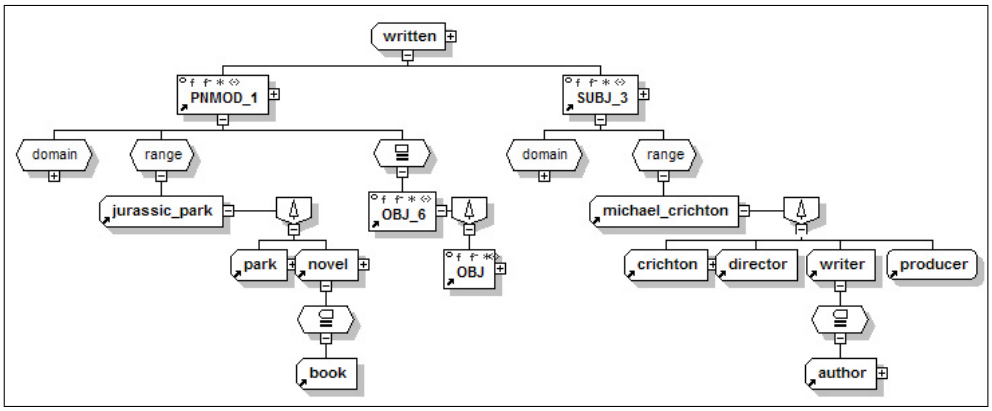

Figure 4: Aligned Ontology-A

2.3 Textual Entailment

The collected information from the alignment phase, both from the classes and properties matching is used to guide us on whether the text entail an hypothesis or not. We have collected such information with the entailment resulted from the previous year RTE data set [Giampiccolo et al., 2007]. The learning data set consisted of 800 text-hypothesis pair with their corresponding 3-way answer, and 800 with 2-way answers. We trained the system using three different machine learning algorithm, that are available from the WEKA package [Witten et al., 1999]. The main classifiers used are the B40 decision tree, nearest neighbor (k=1), and Naive Bayes classifier.

#	Accuracy 2-ways	Accuracy 3-ways	Precision
1	0.688	0.616	0.581
2	0.540	0.520	0.581
3	0.547	0.432	0.581

Table 1: Results at RTE.

3 Performance Evaluation

This year, the textual entailment recognition challenge had two tasks. The 3-way RTE task was concerned in deciding whether the text T entails H, contradicts H, or is unknown from the given T. On the other hand, the 2-way task decided whether T entails H, or not.

We have submitted three runs to the 3-way task. The main difference between the runs is the machine learning algorithm used. The runs for the 3-way task were automatically scored for the 2-way task, were pairs tagged as *"contradiction"* or *"unknown"* are retagged as *"no entailment"*.

Run 1 used the WEKA B40 decision tree classifier, which compared with human annotated answers resulted in 61.6% correct answers for the 3-way classification and 68.8% for 2-ways.

Run 2 used a nearest neighbor classifier (k=1) and resulted in 52% for 3-ways and 54% for 2-ways.

Run 3 used a Naive Bayes classifier and that yielded a score of 43.2% for 3-ways and 54.7% for 2-ways.

An Average Precision score was also used based on the confidence output of the ranked results. Our ranking system simply orders the result by the highest number of directly available classes and properties, which resulted in one score for the three runs of 58.1%. The average precision is computed as the average of the system's precision values at all points for which the human gold standard is *"entailment"*.

Table 1 shows a summary of the results described above.

4 Conclusion and Future Work

In this paper, we have presented the system that we used for recognizing textual entailment. For its first participation, the system achieved acceptable results. On its best run, the system achieved a score of 68% for the two-way task, by simply classifying the ontology alignment output using a decision tree classifier. However, many improvements are needed to achieve better results. By studying our results carefully we have noticed that the performance was significantly affected by longer text. In addition, more training data is needed to improve the classifier results. So our future work will mainly focus on these two aspects.

References

[Giampiccolo et al., 2007] Giampiccolo, D., B. Magnini, I. Dagan, and B. Dolan, 2007. The Third PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.

[Lin, 1998] Lin, D., 1998. Dependency-based evaluation of MINIPAR. *Workshop on the Evaluation of Parsing Systems*:317–330. Granada, Spain, 1998.

[Pedersen et al., 2004] Pedersen, T., S. Patwardhan, and J. Michelizzi, 2004. WordNet::Similarity-Measuring the Relatedness of Concepts. *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*.

[Sibini and Kosseim, 2008] Sibini, Reda and Leila Kosseim, 2008. Rodeo: Reasoning over dependencies extracted online. In European Language Resources Association (ELRA) (ed.), *Proceedings of the The 4th Web as Corpus: Can we do better than Google?, a workshop of the Sixth International Language Resources and Evaluation (LREC'08)*. Marrakech, Morocco.

[Witten et al., 1999] Witten, I.H., University of Waikato, and Dept. of Computer Science, 1999. *Weka: Practical Machine Learning Tools and Techniques with Java Implementations*. Dept. of Computer Science, University of Waikato.