Springer Link

Search

International Conference on Intelligent Text Processing and Computational Linguistics
CICLing 2009: Computational Linguistics and Intelligent Text Processing pp 245-255 | Cite as

AORTE for Recognizing Textual Entailment

Authors | Authors and affiliations

Reda Siblini, Leila Kosseim

Part of the Lecture Notes in Computer Science book series (LNCS, volume 5449)

Abstract

In this paper we
AORTE allows the automatic acquisition and alignment of ontologies from text. The
ted from aligning ontologies created from text fragments is used in classifying

AORTE for Recognizing Textual Entailment

Reda Siblini and Leila Kosseim

CLaC laboratory
Department of Computer Science and Software Engineering
1400 de Maisonneuve Blvd. West
Montreal, Quebec, Canada H3G 1M8
r_sibl@cse.concordia.ca, kosseim@cse.concordia.ca

Abstract. In this paper we present the use of the AORTE system in recognizing textual entailment. AORTE allows the automatic acquisition and alignment of ontologies from text. The information resulted from aligning ontologies created from text fragments is used in classifying textual entailment. We further introduce the set of features used in classifying textual entailment. At the TAC RTE4 challenge the system evaluation yielded an accuracy of 68% on the two-way task, and 61% on the three way task using a simple decision tree classifier.

1 Introduction

In this paper we present a novel method of recognizing textual entailment. Textual entailment is defined as *"a relationship between a coherent text T and a language expression, which is considered as a hypothesis, H. We say that T entails H (H is a consequent of T), if the meaning of H, as interpreted in the context of T, can be inferred from the meaning of T"* [1].
For example, the text:
(T): *Jurassic Park is a novel written by Michael Crichton.*
Entails the following hypothesis (among others):
(H1): *Michael Crichton is an author.*
(H2): *Jurassic Park is a book.*
(H3): *Michael Crichton is the author of the book Jurassic Park.*

Recognizing textual entailment is a fundamental task to many applications in natural language processing, such as in *Information Retrieval* where retrieving relevant documents could be seen as finding documents containing the text that entails the information we are looking for, in *Information Extraction* where the extraction of information is based on a set of templates that entail the information that we would like to extract, in *Question Answering* where candidate answers are snippets that entail the question we want to answer, and in *Summarization* where redundancy can be avoided by detecting textual entailment.

The remainder of the paper is organized as follows. Section 2 presents related work in recognizing textual entailment. In Section 3 we give an overview of our approach, focusing on the main features that were used to classify textual entailment. Section 4 presents a performance evaluation of our system at the recent TAC RTE-4 2008 challenge, and finally in Section 5, we present our conclusion.

2 Related Work

Current methods for recognizing textual entailment are based on a measure of similarity between the text T and the hypothesis H. These methods can be categorized into three main approaches: The first and most popular approach is based on a different set of similarity matching techniques that usually differ by the assumption they make for measuring the similarity. For example, some calculate a similarity measure assuming word independence such as the system of [2], others assume some sort of relationship between words such as the use of parse trees as in the system of [3]. In addition to defining a similarity measure, these methods usually rely heavily on the use of machine learning techniques to classify textual entailment.

The second type of approach is more of a traditional one that relies on a logic based semantic representation of the text and a theorem prover to prove the hypothesis, such as the COGEX system of [4].

The last type of approach is a combination of the two first categories, such as the system of [5] that also relies greatly on world knowledge. This system has achieved the best results on the Recognizing Textual Entailment challenge for the last three years.

The method we describe here can be categorized in the last group. It is a novel approach that relies on knowledge representation, use of extracted world knowledge from the web, and machine learning. But what is unique about it is its use of available techniques in acquiring and aligning ontologies, in addition to machine learning in order to classify textual entailment. The next section will explain our approach in more details.

3 The AORTE Approach

Our approach for recognizing textual entailment is based on the automatic acquisition of an ontology from the text T, and another ontology from the hypothesis H, and the alignment of the acquired ontologies. The textual entailment problem is then reduced to a classification one based on the resulted aligned ontology. In this paper we will not present the details of the creation and alignment of the ontologies, but rather will focus on the classification of textual entailment based on the aligned ontologies.

3.1 Ontology Acquisition

Figure 1 shows a diagram of AORTE's system architecture. As shown in the Figure, given a Text and a Hypothesis, the system automatically acquires an ontology from each, namely ontology-T and ontology-H.

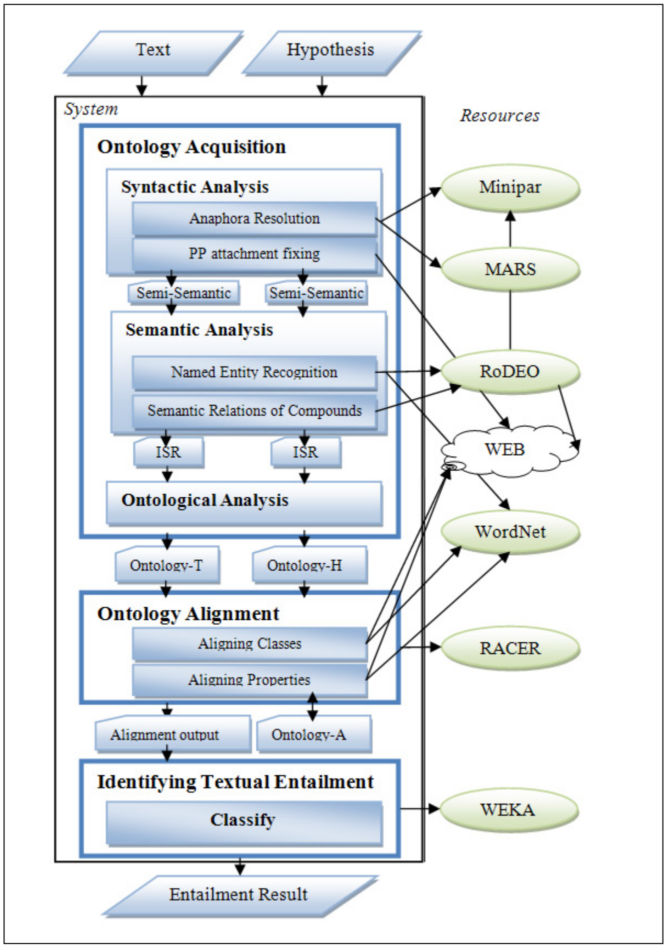

Fig. 1. AORTE System Architecture

The ontology acquisition phase includes different steps that support the acquisition of classes, properties, and instances of an ontology. Briefly, the first step is based on syntactic analysis, which uses the Minipar dependency parser [6], the MARS anaphora resolution system [7], and a set of transformation rules based on part of speech tags to create a structure that is referred to as a semi-semantic structure. The second step to ontology acquisition uses the semi-semantic structure to create a semantic one based on a set of transformation rules and restrictions, in addition to RoDEO's named entities and noun compounds semantic relation extractor [8]. The semantic structure is a semantic underspecified representation of a sentence described by a set of ternary predicate argument relations, characterized by having a predicate as a property that is always relating a governing verb to a content word. For example, the sentence *"Michael*

Crichton is the author of Jurassic Park" could be represented as a set of *"(Predicate(Governing Verb, Content Word[Type])"* as follows: *"writer(write, Michael Crichton[author]) & writable(write, Jurassic Park[book])"*. RoDEO is responsible for adding relevant world knowledge that is extracted from the web and added to the ontologies, such as the type of content word as in *"book"* for the type of *"Jurassic Park"* in our example, or the verb relating compound nouns as in *"write"* relating *"author"* to *"book"*. The last step in ontology acquisition is the ontological analysis which transforms the created semantic structures into a formal semantic representation expressed in the Ontology Web Language (OWL) and more specifically in OWL-DL, a subset of OWL supporting a decidable (SHOIN(D)) description logic. OWL is a semantic markup language for defining and instantiating web ontologies, and it is based on description logic. It is a vocabulary extension of RDF (the Resource Description Framework), derived from the DAML+OIL (DARPA Agent Markup Language and Ontology Interchange Language), and based on XML (Extensible Markup Language). An OWL ontology may include descriptions of classes, properties (relations between two classes), the classes instances, and a set of axioms. The main reason for selecting OWL is the existence of well studied reasoners that can be used to reason over the created knowledge base schema and instances, the availability of well studied powerful query formalism, the possibility of applying rules to the created knowledge base using backward or forward chaining, and the ability to integrate multiple knowledge bases. The reasoner that we are using is RACER [9]. The RACER system (an acronym for Renamed ABox and Concept Expression Reasoner) is a reasoner that implements tableau calculus for description logic (DL) and supports the web ontology languages DAML+OIL, RDF, and OWL.

In order to illustrate the system's output at each stage, let us take the following example from the Recognizing Textual Entailment (RTE3) challenge development set [10].

(T): *Jurassic Park is a novel written by Michael Crichton.*

(H): *Michael Crichton is the author of the book Jurassic park.*

Figure 2 shows a graphical representation of the automatically created ontology from T (ontology-T); while Figure 3 shows the ontology created from H (ontology-H). In these graphs, rectangles represent classes, solid arrows represent subclasses, ovals represent properties, dotted lines represent property domain, and dotted arrows represent property range. What is particular about these automatically created ontologies is that they are fine-grained in the sense that almost every occurrence of a content word in the text results in the creation of a class, every verb semantic role results in the creation of a property, and additional world knowledge extracted from the web is also added.

As we can see in Figure 2, the created properties are the arguments of the verb *Write*, where it has two properties: WRITER having as range the class *Michael Crichton*, and a WRITABLE having as range the class *Jurassic Park*. In addition the ontology includes a taxonomy created from the extracted types of named entities, and the syntactic structure of the sentence, where *Jurassic Park* is a subclass of *Novel* and *Michael Crichton* is a subclass of *Writer, Director, and*

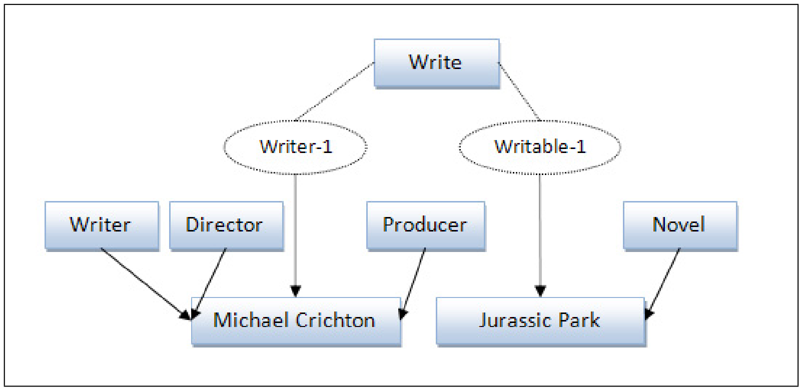

Fig. 2. Example of ontology-T for the sentence *Jurassic Park is a novel written by Michael Crichton.*

Producer. It should be noted that our definition of an instance as a representation of a snapshot of the world at a certain time, would force us to consider a named entity, such as *Michael Crichton*, as a class containing many instances that each describes a different situation of the class at a different time.

In Figure 3 the class *Write*, which was not actually explicitly mentioned in the hypothesis text, was added in the ontology using the RoDEO system that extracted it from the web as a verb that characterizes the relationship between an *Author* and a *Book*. The arguments of the verb *Write* are then added as properties of this class, were the WRITER property takes as range *Michael Crichton*, and the WRITABLE property takes as range *Jurassic park*.

You can notice from these two graphs that these ontologies are quite similar to each other. Only the two classes: *Author* and *Book*, which are available in the ontology-H, are missing from the ontology-T.

3.2 Ontology Alignment

The next stage of the AORTE system as shown in Figure 1, is the alignment of the created ontologies to form a single ontology, namely ontology-A. The alignment phase aligns the classes and properties of the two created ontologies. The alignment takes as its base the ontology created from T and adds to it the classes and properties that align from the hypothesis ontology. This stage is based on our implementation of the S-match algorithm [11] that uses the verbOcean lexical patterns [12] in addition to WordNet [13] to perform the semantic alignment of classes and properties.

Figure 4 shows a graphical representation of the aligned ontology example, where the arrows represent an equivalence axiom between classes or properties. Note that in this specific example, all the classes and properties have been aligned in the resulted ontology. This is the basis of our hypothesis for recognizing textual

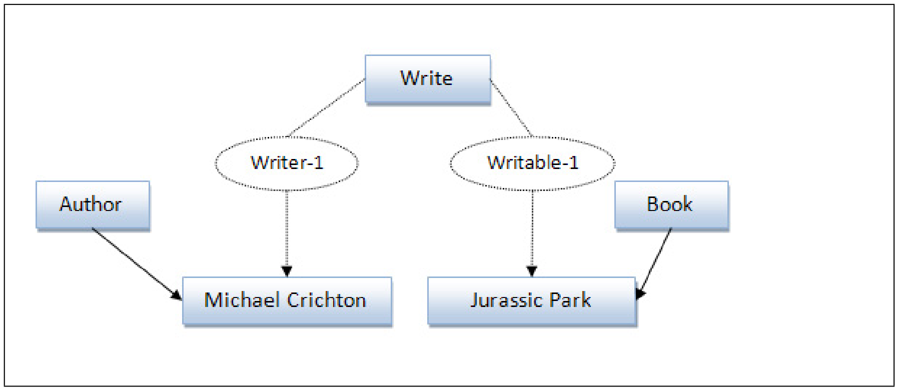

Fig. 3. Example of ontology-H for the sentence *Michael Crichton is the author of the book Jurassic park.*

entailment, where we take the resulted alignments as features for classifying textual entailment.

3.3 Identifying Textual Entailment

Our hypothesis for recognizing textual entailment is that if a high proportion of classes and properties can be aligned between the two created ontologies, then most probably we have an entailment. Consequently, we created a set of features based on the aligned ontology that we believe may be helpful in classifying textual entailment.

The features are:

F1: Available Classes This feature represents the percentage of the classes in ontology-H that were available in ontology-T.

F2: Available Properties This feature represents the percentage of the properties in ontology-H that were available in ontology-T.

F3: Available Sub-Classes The percentage of subclass relationships between classes in ontology-H that are available in the ontology-A.

F4: Equivalent Classes The percentage of equivalent classes available in ontology-A from the number of classes in ontology-H; where an equivalent class is a class having a synonym relation discovered using Wordnet.

F5: Possible Equivalent Classes The percentage of possible equivalent classes available in ontology-A from the number of classes in ontology-H; where a possible equivalent class is an equivalent class that has been labeled by AORTE as being a synonym extracted from the web using the verbOcean lexical patterns.

F6: Equivalent Properties The percentage of equivalent properties available in ontology-A from the number of properties in ontology-H; where an equivalent property is a property having a synonym relation discovered using Wordnet.

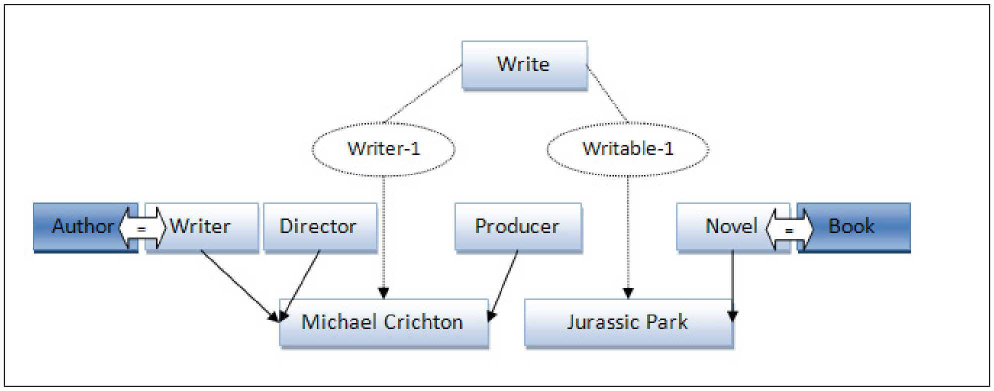

Fig. 4. Example of ontology-A, the alignment of ontology-T of Fig. 2 and ontology-H of Fig. 3

F7: Possible Equivalent Properties The percentage of possible equivalent properties available in ontology-A from the number of properties in ontology-H; where a possible equivalent properties is an equivalent property that has been labeled by AORTE as being a synonym extracted from the web using the verbOcean lexical patterns.

F8: Disjoint Classes The percentage of disjoint classes available in ontology-A from the number of classes in ontology-H. Disjoint classes in OWL are two classes that do not have members in common, in our case it means that the two classes representing content words are antonyms, these antonym relations are discovered using Wordnet.

F9: Possible Disjoint Classes The percentage of possible disjoint classes available in ontology-A from number of classes in ontology-H; where a possible disjoint class is a disjoint class that has been labeled by AORTE as being an antonym extracted from the web using the verbOcean lexical patterns.

F10: Disjoint Properties The percentage of disjoint properties available in ontology-A from the number of properties in ontology-H. Similar to disjoint classes, that represents content words that are antonyms, and these antonym relations are discovered using Wordnet.

To better illustrate how these features are computed, let us take our examples of ontology-T, ontology-H and ontology-A as shown in Figures 2, 3, and 4. Table 1 shows the necessary parameters needed to compute our feature values. These parameters are retrieved by querying ontology-T, ontology-H, and ontology-A. The querying should not be seen as a simple matching of classes in a repository but as a logical inference that is performed using an inference engine. for example, if we queried the ontology-A for if *Michael Crichton is an author*, the reasoner will return *True* and this will be added to the number of subclass relations in ontology-A parameter. And with these parameters, we can now compute our features:

Parameter	Value	Parameter	Value
Classes in ontology-T	7	Properties in ontology-T	2
Subclass relationships in ontology-T	4	Classes in ontology-H	5
Properties in ontology-H	2	Subclass relationships in ontology-T	2
Classes in ontology-A	9	Properties in ontology-A	2
Subclass relationships in ontology-A	6	Equivalent classes in ontology-A	2
Equivalent properties in ontology-A	0	Possible equivalent classes in ontology-A	0
Possible equivalent properties in ontology-A	0	Disjoint classes in ontology-A	0
Possible disjoint classes in ontology-A	0	Possible disjoint classes in ontology-A	0
Disjoint properties in ontology-A	0		

Table 1. Parameters from ontology-T, ontology-H, and ontology-A used to compute the feature values

$$
\begin{array}{ll}
\mathrm{F1} = (7+5-9)/5 = 0.6 & \mathrm{F2} = (2+2-2)/2 = 1 \\
\mathrm{F3} = (4+2)/6 = 1 & \mathrm{F4} = 2/5 = 0.4 \\
\mathrm{F5} = 0 & \mathrm{F6} = 0 \\
\mathrm{F7} = 0 & \mathrm{F8} = 0 \\
\mathrm{F9} = 0 & \mathrm{F10} = 0
\end{array}
$$

We used three classifying algorithms: the B40 decision tree classifier based on ID3, a k-Nearest neighbor classifier with k=1, and a Naïve Bayes classifier. We used as training set the 800 text-hypothesis pairs from the RTE3 pilot task dataset [10]. The RTE3 pilot task is the task of recognizing textual entailment; where the dataset is annotated into three decisions: *yes* for entailment, *no* for no entailment, and *unknown*. We ran the AORTE ontology acquisition and alignment to create ontology-T, ontology-H and ontology-A for all 800 pairs. For each pair, we then extracted the 10 features described above, then trained the classifiers. Table 2 shows a sample from the training set.

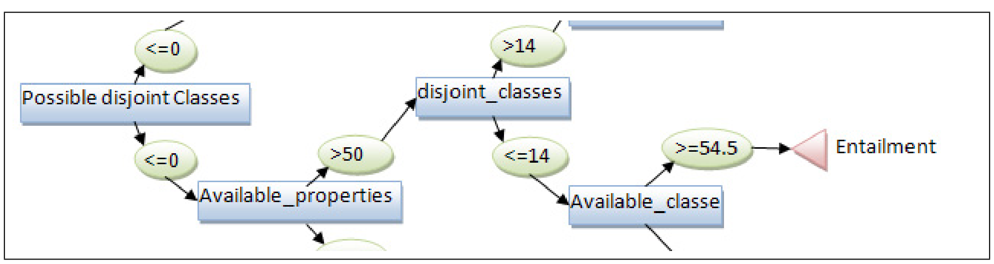

Fig. 5. Part of the decision tree learned from the ontology alignment of RTE3 data set showing the features decision nodes represented by rectangles, each followed by the chance nodes represented by ovals, and ending with a triangle followed by the related decision.

Text	Hypothesis	F1	F2	...	F10	Entailment
The sale was made to pay Yukos' US$ 27.5 billion tax bill, Yuganskneftegaz was originally sold for US$ 9.4 billion to a little known company Baikalfinansgroup which was later bought by the Russian state-owned oil company Rosneft.	Baikalfinansgroup was sold to Rosneft.	1	0		0	YES
A decision to allow the exiled Italian royal family to return to Italy may be granted amid the discovery that the head of the family, Prince Vittorio Emmanuele, addressed the president of Italy properly. He has called President Ciampi "our president, the president of all Italians".	Italian royal family returns home.	0.56	0.6		0	NO
Amsterdam police said Wednesday that they have recovered stolen lithographs by the late U.S. pop artist Andy Warhol worth more than $1 million.	Police recovered 81 Andy Warhol lithographs.	0.77	0.2		0	UNKNOWN

Table 2. Sample of annotated Text-Hypothesis pairs from RTE3

Figure 5) shows part of the decision tree learned from our training set that is relevant to our T-H example. An analysis of the full decision tree indicates that the most discriminating features are the following:

1. F9: Possible disjoint classes (root of the tree).
2. F5, F2, and F3: Possible equivalent classes, available properties, and available sub-classes.
3. F1, F4 and F8: Available classes, equivalent classes and disjoint classes.
4. F6 and F7: Equivalent properties and possible equivalent properties.
5. F10: Disjoint properties.

Traversing the relevant part of the learned decision tree (shown in Figure 5) shows that for our specific example, the decision tree would classify the relation as entailment.

4 Performance Evaluation

To evaluate our system, we participated in the recent TAC-RTE challenge. The Text Analysis Conference (TAC) is a conference that comprises a collection of tracks concerned with the evaluation of different NLP related technologies. One of these tracks is the Recognizing Textual Entailment (RTE) Track, which has the goal of providing a framework to systems that recognize when text entails another. The RTE challenge define textual entailment as *"a directional relationship between two text fragments", where a Text (T) entails a Hypothesis (H) if* *the truth of H can be inferred from T within the context induced by T"*. The RTE challenge is also divided into two tasks: A three way task in which a system must classify textual entailment into: *"Entailment"* if T entails H, a *"Contradiction"* if T contradicts H and *"Unknown"* if the system cannot determine the truth of H based on T. A two way task in which a system must classify textual entailment into: *"Entailment"* if T entails H, and *"No entailment"* if T does not entail H. In order to evaluate our AORTE system, we used the RTE4 challenge, and classified the given 1000 T-H pair into the three way task (*Entailment, Contradiction, and Unknown*). The evaluation is done automatically, where the classifications returned by a system are compared to human annotated golden standard, and the returned score is the accuracy or the percentage of matching judgments. As the RTE4 task did not provide a development set, we used the RTE3 pilot dataset introduced in the previous section for training.

We submitted three different runs, each for a different classification algorithm. In the first run we used the B40 decision tree classifier, and compared to human annotated answers, this run resulted in an accuracy of 61.6%. For the second run we used the nearest neighbor classifier (k=1) and resulted in an accuracy of 52%. In the last run we employed a Naïve Bayes classifier that yielded a score of 43.2%. The best run that scored 61.6% was ranked 2nd when compared to other system that participated in the same challenge.

The RTE challenge automatically converts the three way submitted runs of each system into two way runs by automatically conflating *"Contradiction"* and *"Unknown"* to *"No Entailment"*. The B40 decision tree classifier on the two way run scored a 68.8%, the nearest neighbor classifier achieved 54%, and the Naive Bayes classifier marked a 54.7%. The best run of 68.8% was ranked 2nd when compared to other system that participated in the 3-way challenge and had their answers automatically converted to the two-way format.

In addition to the accuracy measure, the challenge provided the possibility of ranking the textual entailment pairs by confidence. Where the more confident the system is in an entailment the higher it is placed or ranked in the evaluated set. This score is labeled an average precision score, where it is computed as the average of the systems precision values at all points in the ranked list in which the gold standard annotation is *"Entailment"*. As average precision is relevant only for a binary annotation, in the case of three-way judgment submissions the pairs tagged as *"Contradiction"* and *"Unknown"* will be conflated and re-tagged as *"No entailment"*.

We ranked the resulted classification in our system simply by highest number of available classes and properties. So the highest is the percentage of available classes + available properties in the aligned ontologies of a T-H pair having an *"Entailment"* result, the higher it is placed on the result set. As such, the system had an average precision of 58.1% for all runs.

5 Conclusion

This paper proposed a textual entailment recognizing approach based on the alignment of ontologies that are automatically extracted from text. The approach classifies textual entailment by learning from the overlap of classes and properties between the text ontologies and the hypothesis ontology, the percentage of equivalent and disjoint classes and properties in the aligned ontology, and other ontology-alignment related features. The system performance was evaluated using the Recognizing Textual Entailment (RTE-4) challenge, resulting in a accuracy of 68% on the two-way task, and 61% on the three way task for the best run that uses a simple decision tree classifier, which ranked 2nd when compared to the other systems that participated in the challange. By carefully studying our results we have realized that the system performance had significantly been affected by the text length, as a result our future work will focus on resolving this issue and mainly on improving the association of relevant knowledge. In addition, we will work on providing a detailed analyze of the type of textual entailment the system can handle and specifically the contribution of each of the system's component to the overall performance.

References

[1] Dagan, I., Glickman, O.: Probabilistic textual entailment: Generic applied modeling of language variability. Learning Methods for Text Understanding and Mining (2004)

[2] Corley, C., Csomai, A., Mihalcea, R.: A Knowledge-based Approach to Text-to-Text Similarity. Recent Advances in Natural Language Processing IV: Selected Papers from RANLP 2005 (2007)

[3] Kouylekov, M., Magnini, B.: Tree Edit Distance for Textual Entailment. Recent Advances in Natural Language Processing IV: Selected Papers from RANLP 2005 (2007)

[4] Tatu, M., Iles, B., Moldovan, D.: Automatic Answer Validation Using COGEX. Lecture Notes in Computer Science **4730** (2007)

[5] Hickl, A., Bensley, J.: A Discourse Commitment-Based Framework for Recognizing Textual Entailment. ACL (2007)

[6] Lin, D.: Dependency-based evaluation of Minipar. Workshop on the Evaluation of Parsing Systems (1998) Granada, Spain, 1998.

[7] Mitkov, R., Evans, R., Constantin, O.: A new, fully automatic version of Mitkov's knowledge-poor pronoun resolution method. In: Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2002). (2002) Mexico City, Mexico.

[8] Sibini, R., Kosseim, L.: Rodeo: Reasoning over dependencies extracted online. In: 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 (2008)

[9] Haarslev, V., Moller, R.: Racer: A Core Inference Engine for the Semantic Web. Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools (2003) Sanibel Island, Fl, USA, 2003.

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

[11] Giunchiglia, F., Shvaiko, P., Yatskevich, M.: S-Match: an Algorithm and an Implementation of Semantic Matching. The Semantic Web: Research and Applications: First European Semantic Web Symposium, ESWS 2004, Heraklion, Crete, Greece, May 10-12, 2004: Proceedings (2004)

[12] Chklovski, T., Pantel, P.: VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In: Proc. of EMNLP-04. Barcelona, Spain. (2004)

[13] Fellbaum, C.: WordNet: an electronic lexical database. MIT Press USA (1998)