Using Shallow Semantic Parsing and Relation Extraction for Finding Contradiction in Text

Minh Quang Nhat Pham, Minh Le Nguyen and Akira Shimazu
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa, 923-1292, JAPAN
{minhpqn,nguyenml,shimazu}@jaist.ac.jp

Abstract

Finding contradiction text is a fundamental problem in natural language understanding. Previous work on finding contradiction in text incorporate information derived from predicate-argument structures as features in supervised machine learning frameworks. In contrast to previous work, we combine shallow semantic representations derived from semantic role labeling with binary relations extracted from sentences in a rule-based framework. Evaluation experiments conducted on standard data sets indicated that our system achieves better recall and F1 score for contradiction detection than most of baseline methods, and the same recall as a state of the art supervised method for the task.

1 Introduction

Contradiction detection (CD) in text is a fundamental task in natural language understanding, and necessary for many applications (De Marneffe et al., 2008; Voorhees, 2008). For instance, contradictions need to be recognized by question answering systems or multi-document summarization systems (Harabagiu et al., 2006). The task is to detect whether the contradiction relationship exists in a pair of a text *T* and a hypothesis *H*.

There are several approaches to the CD task. Contradiction detection can be formalized as a binary classification problem (Harabagiu et al., 2006; De Marneffe et al., 2008). The main effort of work which adopt this approach is to find out effective features for recognizing contradiction. The other approach is using functional relations indicated by verb or noun phrases for detecting contradiction (Ritter et al., 2008).

Beyond string-based matching approaches, one can approach to the CD task by applying logical inference techniques. Although the logical inference approach may obtain good precision, it is not widely used for the task due to the fact that full predicate-logic analysis is currently not practical for wide-coverage semantic processing (Burchardt et al., 2009). Given that fact, (Burchardt et al., 2009) pointed out that using shallow semantic representations based on predicate-argument structures and frame knowledge is an intuitive and straightforward approach to textual inference tasks.

In contrast to previous work which integrate predicate-argument structures as features in machine learning-based systems (Harabagiu et al., 2006; De Marneffe et al., 2008), this paper combines shallow semantic representations derived from semantic role labeling with binary relations extracted from sentences for the CD task. The proposed system consists of two modules. The first module relies on the alignment of semantic role (SRL) frames extracted from the text and the hypothesis in each pair while the second one performs contradiction detection over binary relations extracted from the pair. If the SRL-based module fails to identify the contradiction relationship in the pair, the second module will be applied. We expect that the second module will improve the coverage of the first one. Evaluation experiments on standard data sets obtained from RTE challenges (Giampiccolo et al., 2007; Giampiccolo et al., 2008; Bentivogli et al., 2009) show that the proposed system achieves better recall and F1 score for contradiction detection than most of baseline methods, and the same recall as a state of the art supervised method for the task.

2 Linguistic Analysis

After parsing the text and the hypothesis of a pair by using Stanford CoreNLP [1], we utilize SENNA

[1] Stanford CoreNLP is available online on: http://nlp.stanford.edu/software/corenlp.shtml

International Joint Conference on Natural Language Processing, pages 1017–1021,
Nagoya, Japan, 14-18 October 2013.

package[2] (Collobert et al., 2011) for semantic role labeling. Then, we extract SRL frames from the output of SENNA. An SRL frame consists of a verb predicate and a list of SRL elements.

In the system, we use REVERB (Fader et al., 2011) – a tool which can automatically identify and extract binary relations from English sentences. The input of REVERB is a POS-tagged and NP-chunked sentence and its output is a set of extraction triples of the form (*arg1*, *R*, *arg2*), in which *R* represents the relation phrase between two arguments: *arg1* and *arg2*.

REVERB cannot extract some useful relations such as “isA” relations which specify the equivalent relation of two objects. In addition, in some cases, relation phrases of two extraction triples cannot be compared without using inference rules that specify the entailment relationship between two triples. Therefore, we propose several simple heuristic methods to extract additional binary relations from a text segment.

First, we extract “isA” relations from three information sources: i) co-reference resolution information; ii) noun phrases which the ending parts are recognized as a named entity;; and iii) “abbrev” relations in dependency parses.

Second, entailment rules or inference rules which specify directional entailment relations between two text fragments have been shown to be useful for RTE and question answering (Berant et al., 2011). In this study, we transform triples generated by REVERB by looking up the corpus of $30,000$ entailment rules between typed predicates obtained from (Berant et al., 2011).

3 Contradiction Detection by Matching Semantic Frames

Let us denote an SRL frame by a tuple $S = \{V, E_1, \ldots, E_k\}$, where V is used to denote the verb predicate; and E_i represents the i-th SRL element in the frame. Each SRL element has a type and underlying words. Types of SRL elements follow the annotation guideline in PropBank (Palmer et al., 2005). SRL elements can be arguments or modifiers (adjuncts). We denote two sets of SRL frames of *T* and *H* by $T = \{S_i^{(t)}\}_{i=1}^m$ and $H = \{S_j^{(h)}\}_{j=1}^n$, in which m and n are the number of SRL frames extracted from *T* and *H*, respectively.

3.1 Contradiction Detection Model

The contradiction detection model consists of a contradiction function $\mathcal{F}_S(T, H)$ which calculates the contradiction measurement for the pair (*T*, *H*) on their SRL frames. Then, $\mathcal{F}_S(T, H)$ is compared with a threshold value t_1. If $\mathcal{F}_S(T, H) \geq t_1$, we determine that *T* and *H* are contradictory.

In order to define the contradiction function $\mathcal{F}_S(T, H)$, we rely on the assumption that *T* and *H* are contradictory if there exists an event indicated by an SRL frame in *H*, which is incompatible with an event indicated by *T*. Formally, the function $\mathcal{F}_S(T, H)$ is defined as following:

$$\mathcal{F}_S(T, H) = \max_{S_i^{(t)} \in T, S_j^{(h)} \in H} \mathbf{f}(S_i^{(t)}, S_j^{(h)}), \quad (1)$$

where $S_i^{(t)}$ and $S_j^{(h)}$ are two SRL frames in *T* and *H*, respectively; and $\mathbf{f}(S_i^{(t)}, S_j^{(h)})$ is a contradiction function defined on the two SRL frames.

Next, we define the function $\mathbf{f}(S_1^{(t)}, S_2^{(h)})$ of two SRL frames $S_1^{(t)} \in T$ and $S_2^{(t)} \in H$. For concreteness, we denote $S_1^{(t)} = \{V_1, E_1^{(1)}, \ldots, E_k^{(1)}\}$ and $S_2^{(h)} = \{V_2, E_1^{(2)}, \ldots, E_\ell^{(2)}\}$.

The function $\mathbf{f}(S_1^{(t)}, S_2^{(h)})$ relies on the alignment of SRL elements across two frames. Since the number of SRL elements in an SRL frame is not very large, we propose a greedy alignment algorithm that considers all possible pairs of an SRL element in $S_1^{(t)}$ and an SRL element in $S_2^{(h)}$. The core part of the greedy algorithm is the similarity measure between two SRL elements. We apply the local lexical level matching method (Dagan et al., 2007) to calculate the similarity of two SRL elements. In addition, we utilize co-reference resolution information by substituting mentions found in an SRL element with their equivalent mentions in the corresponding co-reference chain.

After generating the alignment between elements of two SRL frames, we define the contradiction function $\mathbf{f}(S_1^{(t)}, S_2^{(h)})$ as follows.

From the rationale that two events are not contradictory if they are not related, we filter out “not contradictory” SRL frame pairs by calculating their relatedness. The relatedness of two SRL frames is defined as product of the relatedness of their verb predicates and SRL elements:

$$R(S_1^{(t)}, S_2^{(h)}) = R(V_1, V_2) \times max_{i,j} R(E_i^{(1)}, E_j^{(2)}), \quad (2)$$

[2] SENNA is available online on: http://ml.nec-labs.com/senna/

where R represents the relatedness between two items; $E_i^{(1)} \in S_1^{(t)}$ and $E_j^{(2)} \in S_2^{(h)}$ are SRL elements; V_1 and V_2 are verbs of $S_1^{(t)}$ and $S_2^{(h)}$, respectively.

The relatedness of two verbs is assigned to 1.0 if their relation is found in WordNet (Fellbaum, 1998) or in VerbOcean database (Chklovski and Pantel, 2004). In other cases, we employ WordNet::Similarity package (Pedersen et al., 2004) to compute the similarity of two verbs. The relatedness of two SRL elements $E_i^{(1)}$ and $E_j^{(2)}$ is defined as the local lexical level matching score.

The relatedness of two SRL frames is compared with a threshold. If it is below the threshold, then $S_1^{(t)}$ and $S_2^{(h)}$ are not related.

If two SRL frames are related, we consider two situations: 1) two verb predicates are matching and 2) Two verb predicates are opposite. Note that if two verb predicates are neither matching nor opposite, $\mathbf{f}(S_1^{(t)}, S_2^{(h)})$ is also assigned to 0.

In the system, that two verbs are matching are determined by utilizing synonyms in WordNet and WordNet-base semantic similarity. If two verb are matching, the function $\mathbf{f}(S_1^{(t)}, S_2^{(h)})$ is defined based on the alignment generated in the alignment process. We use the incompatibility of aligned arguments and modifiers such as temporal, location, or negation modifiers to calculate $\mathbf{f}(S_1^{(t)}, S_2^{(h)})$.

In the second case, two verbs are opposite if they are found as antonym verbs in WordNet or opposite verbs in VerbOcean. In this case, the contradiction function $\mathbf{f}(S_1^{(t)}, S_2^{(h)})$ is defined as the similarity of their SRL elements. We define the element-based similarity of two frames as the product of similarity scores of the aligned elements having the same type.

4 Contradiction Detection by Relation Matching

The main idea of this module is as follows. In the first step, we extract triples from *T* and *H* by using REVERB tool and our heuristics. Next, we compare each triple in *H* with every triple in *T*, and determine whether the contradiction relationship exists in some pairs of triples.

Formally, we denote a extraction triple by (x, r, y) where x and y respectively represent the first and second argument, and r represents the relation phrase of the triple.

We denote $T = \{(x_i^{(t)}, r_i^{(t)}, y_i^{(t)})\}_{i=1}^m$ and $H = \{(x_j^{(h)}, r_j^{(h)}, y_j^{(h)})\}_{j=1}^n$. Here, m and n are respectively the numbers of triples in *T* and *H*. The contradiction detection task is reduced to searching for incompatible triple pairs across *T* and *H*. We define the contradiction function on triples of *T* and *H* as follows.

$$\mathcal{F}_T(T, H) = \max_{T_i \in T; H_j \in H} \mathbf{g}(T_i, H_j), \qquad (3)$$

where T_i is the *i*-th triple of *T*; H_j is the *j*-th triple of *H*; and $\mathbf{g}(T_i, H_j)$ is the contradiction function of the two triples T_i and H_j.

The function $\mathbf{g}(T_i, H_j)$ is based on the mismatch of two triples T_i and H_j. We consider three cases as follows. If their relation phrases and first arguments are matching, the mismatch of second arguments will be calculated. If two relation phrases are matching and roles of arguments in the two triples are exchanged, $\mathbf{g}(T_i, H_j)$ is assigned to 1.0. However, this rule is not applied for "isA" (equivalent) relations. In contrast, if two relation phrases are opposite, the similarity measures of first arguments and second arguments are taken into account.

In the procedure for calculating $\mathbf{g}(T_i, H_j)$, we need to determine whether two relation phrases $r_i^{(t)}$ and $r_j^{(h)}$ are matching or not. If the surface and base forms of two relation phrases are different, we use WordNet to detect whether main verbs of $r_i^{(t)}$ and $r_j^{(h)}$ are synonyms. In order to check if two relation phrases $r_i^{(t)}$ and $r_j^{(h)}$ are opposite or not, we utilize antonym relations in WordNet and opposite relations in VerbOcean.

In the module, that two arguments are matching is checked by using their similarity. The similarity score of two arguments is computed by the same method as that for computing the similarity of two SRL elements. When we detect the contradiction of two arguments, we use the contradiction rule as follows. Two arguments are contradictory if they include two entities having the same type but different values. Especially, we take into account four categories: NUMBER, DATE, TIME, and LOCATION. In other cases, we use the similarity of two arguments as the evidence for contradiction detection.

5 Evaluation Experiments

5.1 Data Sets

In experiments, we evaluate the proposed method on the test sets of the three-way subtask at RTE-

Table 1: Label distribution in three test sets

Data Set	Contradiction	Entailment	Unknown	Total
RTE-3 Test	72	410	318	800
RTE-4 Test	150	500	350	1000
RTE-5 Test	90	300	210	600

3, RTE-4, and RTE-5 competitions (Giampiccolo et al., 2007; Giampiccolo et al., 2008; Bentivogli et al., 2009). The development sets provided at each competition are used to tuned threshold values in two CD modules of the system. The three-way subtask requires participant systems to decide whether the entailment, contradiction, or unknown relationship exists in a pair. Since in this study, we focus on contradiction relationship in a text pair, entailment and unknown labels in data sets are converted into non-contradiction labels. Table 1 provides statistics on the test sets of three-way subtask in RTE-3, RTE-4, and RTE-5.

The data sets used in experiments are unbalanced, so the average accuracy over all labels is not an appropriate evaluation measures. Therefore, we use Precision, Recall, and F1 scores of the contradiction label as evaluation measures.

5.2 Baseline Methods

The first baseline method is the method presented in (De Marneffe et al., 2008), which employed supervised machine learning techniques for the CD task. To the best of our knowledge, (De Marneffe et al., 2008) is the only contradiction detection-focused work that evaluates on data sets of RTE challenges.

The second baseline is the BLUE system of Boeing's team (Clark and Harrison, 2009) at RTE-4 and RTE-5 competitions. The BLUE system adopted the logical inference approach to RTE, which performs inference on logic-based representations of the text and the hypothesis in a pair. We use best scores among submitted runs of the BLUE system at each competition.

In experiments, we also compare the results achieved by our system with average results of submitted systems for three-way subtask at RTE-3, RTE-4 and RTE-5 challenges. The numbers submitted systems in RTE-3, RTE-4 and RTE-5 for the three-way subtask are 12, 34, and 24 submissions, respectively.

In order to assess the effectiveness of the two-stage system scheme, we separately run each CD module on the three data sets and compare the results with those of the combined system.

5.3 Experimental Results

Table 2 provides experimental results achieved on test sets of RTE-3, RTE-4, and RTE-5 challenges by our system and baseline methods. As shown in results, the proposed system consistently obtained better recall values and F1 scores than those of baseline methods except the supervised machine learning-based method in (De Marneffe et al., 2008). Compared with the method presented (De Marneffe et al., 2008), our system achieves the same recall but lower precision.

The results shown in Table 2 indicated that the SRL-based module consistently achieved better recall and F1 score than those of the triple-based module. A possible explanation is that the information contained in shallow semantic representations is richer than that of extraction triples, so the SRL-based module covers more contradiction phenomena than the triple-based module. As expected, the combined system consistently obtained better recall and F1 score than each separate module. Experimental results confirmed our observation that the second backup module increases the coverage of contradiction phenomena for our system.

6 Conclusion

In this paper, we have presented a new rule-based method for finding contradiction in text, which combines shallow semantic representations with binary relations extracted from sentences. We define contradiction measurements on the predicate-argument structures and binary relations extracted from the text and the hypothesis in a pair. We deal with the low-coverage problem of semantic role resources by using a backup module which exploits extraction triples. Experimental results achieved on standard data sets showed that our proposed system obtained better recall and F1 score for contradiction detection than most of baseline methods.

Table 2: Experimental results on three data sets

Method	**RTE-3 Pilot**			**RTE-4 Test**			**RTE-5 Test**		
	P	R	F1	P	R	F1	P	R	F1
De Marneffe (2008)	22.95	19.44	21.04	–	–	–	–	–	–
BLUE system	–	–	–	41.67	10.0	16.13	42.86	6.67	11.54
Average result	10.72	11.69	11.18	25.26	13.47	13.63	26.40	13.70	14.79
SRL-based	13.41	15.27	14.28	22.41	17.33	19.55	22.72	16.67	19.23
Triple-based	22.58	9.72	13.59	26.3	10.0	14.49	19.48	16.67	17.96
Two-stage (our system)	**14.0**	**19.44**	**16.27**	**23.0**	**22.67**	**22.82**	**21.14**	**28.89**	**24.4**

References

L. Bentivogli, I. Dagan, H. T. Dang, D. Giampiccolo, and B. Magnini. 2009. The fifth pascal recognizing textual entailment challenge. In *In Proceedings of TAC Workshop*.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global learning of typed entailment rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 610–619, Portland, Oregon, USA, June. Association for Computational Linguistics.

Aljoscha Burchardt, Marco Pennacchiotti, Stefan Thater, and Manfred Pinkal. 2009. Assessing the impact of frame semantics on textual entailment. *Natural Language Engineering*, 15(Special Issue 04):527–550.

Timothy Chklovski and Patrick Pantel. 2004. Verbocean: Mining the web for fine-grained semantic verb relations. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 33–40, Barcelona, Spain, July. Association for Computational Linguistics.

Peter Clark and Phil Harrison. 2009. Recognizing textual entailment with logical inference. In *In Proceedings of the First Text Analysis Conference (TAC 2008)*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 999888:2493–2537, November.

Ido Dagan, Dan Roth, and Fabio Massimo. 2007. A tutorial on textual entailment.

Marie-catherine De Marneffe, Anna N Rafferty, and Christopher D Manning. 2008. Finding contradictions in text. In *In Proceedings of ACL 2008*.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1535–1545, Stroudsburg, PA, USA. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9.

Danilo Giampiccolo, Hoa Trang Dang, Bernardo Magnini, Ido Dagan, Elena Cabrio, and Bill Dolan. 2008. The fourth pascal recognizing textual entailment challenge. In *In Proceedings of TAC 2008 Workshop*.

Sanda Harabagiu, Andrew Hickl, and Finley Lacatusu. 2006. Negation, contrast, and contradiction in text processing. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06)*.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist.*, 31(1):71–106, March.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet::similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, HLT-NAACL–Demonstrations '04, pages 38–41, Stroudsburg, PA, USA. Association for Computational Linguistics.

Alan Ritter, Stephen Soderland, Doug Downey, and Oren Etzioni. 2008. It's a contradiction – no, it's not: A case study using functional relations. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 11–20, Honolulu, Hawaii, October. Association for Computational Linguistics.

Ellen M. Voorhees. 2008. Contradictions and justifications: Extensions to the textual entailment task. In *Proceedings of ACL-08: HLT*, pages 63–71, Columbus, Ohio, June. Association for Computational Linguistics.