A Proof-Based Annotation Platform of Textual Entailment

Assaf Toledo[1], Stavroula Alexandropoulou[1], Sophie Chesney[2],
Robert Grimm[1], Pepijn Kokke[1], Benno Kruit[3], Kyriaki Neophytou[1],
Antony Nguyen[1], Yoad Winter[1]

[1] - Utrecht University [2] - University College London [3] - University of Amsterdam
{a.toledo,s.alexandropoulou,y.winter}@uu.nl
sophie.chesney.10@ucl.ac.uk, {pepijn.kokke,bennokr}@gmail.com
{r.m.grimm,k.neophytou,a.h.nguyen}@students.uu.nl

Abstract

We introduce a new platform for annotating inferential phenomena in entailment data, buttressed by a formal semantic model and a proof-system that provide immediate verification of the coherency and completeness of the marked annotations. By integrating a web-based user interface, a formal lexicon, a lambda-calculus engine and an off-the-shelf theorem prover, the platform allows human annotators to mark linguistic phenomena in entailment data (pairs made up of a premise and a hypothesis) and to receive immediate feedback whether their annotations are substantiated: for positive entailment pairs, the system searches for a formal logical proof that the hypothesis follows from the premise; for negative pairs, the system verifies that a counter-model can be constructed. This novel approach facilitates the creation of textual entailment corpora with annotations that are sufficiently coherent and complete for recognizing the entailment relation or lack thereof. A corpus of several hundred annotated entailments is currently being compiled based on the platform and will be available for the research community in the foreseeable future.

Keywords: Annotation Platform, Semantic Annotation, Proof System, Formal Model, Textual Entailment, RTE

1. Introduction

The Recognizing Textual Entailment (RTE) corpora (Dagan et al., 2006; Bar Haim et al., 2006; Giampiccolo et al., 2008, a.o) present the challenge of automatically determining whether an entailment relation obtains between a naturally occurring text *T* and a manually composed hypothesis *H*.[1] These corpora, which are currently the only available resources of textual entailments, mark entailment candidates as positive/negative.[2] For example:

Example 1

- T: For their discovery of ulcer-causing bacteria, Australian doctors Robin Warren and Barry Marshall have received the 2005 Nobel Prize in Physiology or Medicine.
- H: Robin Warren was awarded a Nobel Prize.[3]
- Entailment: Positive

However, the linguistic phenomena that underlie entailment in each particular case and their contribution to inferential processes are not indicated in the corpora. In the absence of a gold standard that identifies linguistic phenomena triggering inferences, the inferential processes employed by entailment systems to recognize entailment are not directly accessible and, as a result, cannot be evaluated or improved straightforwardly.

We address this problem through the SemAnTE (Semantic Annotation of Textual Entailment) platform introduced in this paper. The platform allows human annotators to elucidate some of the central inferential processes underlying entailments in the RTE corpus. In 80.65% of the positive pairs in RTE 1–4, annotators found the recognition of entailment to rely on inferences stemming, *inter alia*, from the semantics of appositive, restrictive or intersective modification (Toledo et al., 2013). We decided to focus on the above three phenomena for two reasons. First, they are prevalent in the RTE datasets and, second, their various syntactic expressions can be modeled semantically using a limited set of logical concepts, such as equivalence, inclusion and conjunction.

The annotation platform allows the annotators to mark the above three modification patterns when they are involved in the recognition of entailment by binding the words and constructions in sentences to a lexicon of abstract semantic denotations. The proposed semantic modeling offers an important advantage: it licenses the system to search for formal proofs that substantiate manual annotations and to describe how the modeled phenomena interact and contribute to the recognition process. This is achieved by employing a lambda-calculus engine and a theorem prover.

The platform is currently employed for the preparation of a new corpus of several hundred annotated entailments comprising both positive and negative pairs. In the future, we plan to extend the semantic model to cover other, more complex phenomena.

[1] A short software demonstration paper describing the SemAnTE annotation platform is included in the EACL 2014 proceedings.

[2] Pairs of sentences in RTE 1-3 are categorized in two classes: *yes-* or *no-entailment*; pairs in RTE 4-5 are categorized in three classes: *entailment*, *contradiction* and *unknown*. We label the judgments *yes-entailment* from RTE 1-3 and *entailment* from RTE 4-5 as *positive*, and the other judgments as *negative*.

[3] Pair 222 from the development set of RTE 2.

2. Semantic Model

We model entailment in natural language based on order theory, on a working assumption that entailment describes a *preorder* relation on the set of all possible sentences. Thus, any sentence trivially entails itself (reflexivity); and given two entailments $T_1 \Rightarrow H_1$ and $T_2 \Rightarrow H_2$ where H_1 and T_2 are identical sentences, we assume $T_1 \Rightarrow H_2$ (transitivity). We use a standard model-theoretical extensional semantics, whereby each model M assigns sentences a truth-value in the set $\{0, 1\}$ – the domain of *truth-values* on which we assume the simple *partial order* $\leq$. We adapt Tarski's (1944) theory of truth to entailment relations and consider a theory of entailment adequate if the intuitive entailment preorder on sentences can be described as the pairs of sentences T and H whose truth-values $[\![T]\!]^M$ and $[\![H]\!]^M$ satisfy $[\![T]\!]^M \leq [\![H]\!]^M$ for all models M.

The function of annotations is to link between textual representations in natural language and model-theoretic representations. To this end, the words and structural configurations in T and H are marked with lexical labels that encode semantic meanings for the linguistic phenomena being modeled. These lexical labels are defined formally in a lexicon, as illustrated in Table 1 for major lexical categories over types: e for *entities*, t for *truth-values*, and the functional compounds of e and t.

Category	Type	Example	Denotation
Proper Name	e	Dan	$\mathbf{dan}$
Indef. Article	$(et)(et)$	a	A
Def. Article	$(et)e$	the	ι
Copula	$(et)(et)$	is	IS
Noun	et	bacteria	$\mathbf{bacteria}$
Intrans. verb	et	sit	$\mathbf{sit}$
Trans. verb	eet	receive	$\mathbf{receive}$
Pred. Conj.	$(et)((et)(et))$	and	AND
Res. Adj. (Mod)	$(et)(et)$	short	$R_m(\mathbf{short})$
Res. Adj. (Pred)	et	short	$P_r(\mathbf{short})$
Res. Adj. (Mod)	$(et)(et)$	thin	$R_m(\mathbf{thin})$
Res. Adj. (Pred)	et	thin	$P_r(\mathbf{thin})$
Int. Adj. (Mod))	$(et)(et)$	Dutch	$I_m(\mathbf{dutch})$
Int. Adj. (Pred))	et	Dutch	$\mathbf{dutch}$
Exist. Quant.	$(et)(et)t$	some	SOME

Table 1: Lexicon Illustration

Denotations that are assumed to be arbitrary are given in boldface. For example, the intransitive verb *sit* is assigned the type et, which describes functions from entities to truth-values, and its denotation $\mathbf{sit}$ is an arbitrary function of this type. The denotations of several other lexical items are restricted by the given model M. As illustrated in Figure 1, the coordinator *and* is assigned the type $(et)((et)(et))$, and its denotation is a function that takes a function A of type et and returns a function that takes a function B, also of type et, and returns a function that takes an entity x of type e and returns 1 if and only if x satisfies both A and B.

$$\textsc{a} \;=\; \textsc{is} = \lambda A_{et}.A$$

$$\iota = \lambda A_{et}.\begin{cases} a & A = (\lambda x_e.x = a) \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$\begin{aligned}
\textsc{who}_A \;&=\; \lambda A_{et}.\lambda x_e.\iota(\lambda y.y = x \wedge A(x)) \\
R_m \;&=\; \lambda M_{(et)(et)}.\lambda A_{et}.\lambda x_e.M(A)(x) \wedge A(x) \\
P_r \;&=\; \lambda M_{(et)(et)}.\lambda x_e.M(\lambda y_e.1)(x) \\
\textsc{some} \;&=\lambda A_{et}.\lambda B_{et}.\exists x.A(x) \wedge B(x) \\
\textsc{and} \;&=\lambda A_{et}.\lambda B_{et}.\lambda x_e.A(x) \wedge B(x)
\end{aligned}$$

Figure 1: Functions in the Lexicon

Attaching lexical labels to words and syntactic constructions enables annotators to mark the linguistic phenomena manifested in the data. Moreover, by virtue of its formal foundation, this approach allows annotators to verify that the entailment relation (or lack thereof) that obtains between the textual forms of T and H is also present between their respective semantic forms. This latter step ensures that the annotations provide sufficient information for recognizing the entailment relation in a given pair based on the semantic abstraction. For example, consider the simple entailment *Dan is short and thin* ⇒ *Dan is short* and assume annotations of *Dan* as a proper name, *short* and *thin* as restrictive modifiers in predicate position, and *and* as predicate conjunction. The formal model can be used to verify these annotations by constructing a proof as follows:

For each model M, $[\![\,\textit{Dan [is [short [and thin]]]}\,]\!]^M$

$$\begin{array}{cl|l}
= & (\textsc{is}((\textsc{and}(P_r(\mathbf{thin})))(P_r(\mathbf{short}))))(\mathbf{dan}) & \text{analysis} \\
= & (((\lambda A_{et}.\lambda B_{et}.\lambda x_e.A(x) \;\wedge\; B(x)) & \text{def. of } \textsc{is} \\
 & (P_r(\mathbf{thin})))(P_r(\mathbf{short})))(\mathbf{dan}) & \text{and } \textsc{and} \\
= & P_r(\mathbf{thin})(\mathbf{dan}) \wedge P_r(\mathbf{short})(\mathbf{dan}) & \text{func. app.} \\
\leq & P_r(\mathbf{short})(\mathbf{dan}) & \text{def. of } \wedge \\
= & (\textsc{is}(P_r(\mathbf{short})))(\mathbf{dan}) & \text{def. of } \textsc{is} \\
= & [\![\,\textit{Dan is short}\,]\!]^M & \text{analysis}
\end{array}$$

3. Platform Architecture

The platform's architecture is based on a client-server model, as illustrated in Figure 2.

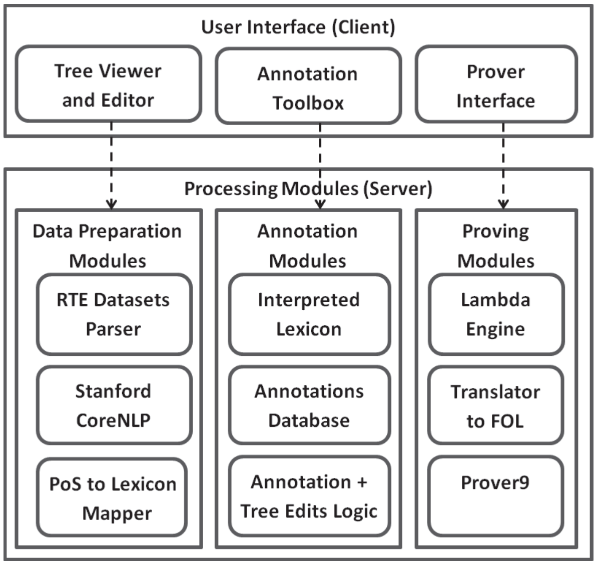

Figure 2: Platform Architecture

The user interface (UI) is implemented as a web-based client using Google Web Toolkit (Olson, 2007) and allows multiple annotators to access the RTE data, to annotate them, and to substantiate their annotations. These operations are done by invoking corresponding remote procedure calls at the server side. We describe the system components as we go over the work-flow of annotating Example 1.

Data Preparation: We extract T-H pairs from the RTE datasets XML files and use the Stanford CoreNLP (Klein and Manning, 2003; Toutanova et al., 2003; de Marneffe et al., 2006) to parse each pair and to annotate it with part-of-speech tags.[4] Subsequently, we apply a naive heuristic to map the PoS tags to the lexicon.[5] This process is performed as part of the platform's installation and when annotators need to simplify the original RTE data in order to avoid syntactic/semantic phenomena that the semantic engine does not support. For example, the fronted *for*-phrase *For their discovery...* is moved after the object of the verb *receive* as fronted adjuncts are not supported. Additionally, the phenomenon of distributivity manifested in the inference *Robin Warren and Barry Marshall have received...* → *Robin Warren has received...*, which is required for recognizing the entailment in this example. We do not model this inference and the construction must therefore be simplified. These simplifications yield T_{simple} and H_{simple} as follows:

- T_{simple}: The Australian doctor Robin Warren has received the great Nobel Prize in Physiology-Medicine for the discovery of the ulcer-causing bacteria.
- H_{simple}: Robin Warren was awarded a Nobel Prize.

Annotation: The annotation is done by marking the tree-leaves with entries from the lexicon. For example, *receives* is annotated as a transitive verb, *ulcer-causing* is annotated as a restrictive modifier (*MR*) of the noun *bacteria*, and *Australian* is annotated as an intersective modifier of the noun *doctors*. In addition, annotators add leaves that mark semantic relations. For instance, a leaf that indicates the apposition between *The Australian doctor* and *Robin Warren* is added and annotated as WHO_A. Furthermore, the annotators fix parsing mistakes as in *the great Nobel Prize in Physiology–Medicine* which was parsed as: [the [great [Nobel Prize]]] [in Physiology–Medicine] and fixed to: [the [great [[Nobel Prize] [in Physiology–Medicine]]]]. The server stores a list of all annotation actions. Figure 3 shows the tree-view, lexicon, prover and annotation history panels in the UI.

Defining Lexical Relations: Our modeling of modification phenomena does not address inferences that rely on lexical knowledge, as in: "Robin Warren has received a prize" → "Robin Warren was awarded a prize". Such lexical relations between the text and hypothesis are marked by the annotators and translated into logical formulas by the proof-system.

Proving: Once all leaves are annotated and the tree structures of T_{simple} and H_{simple} are manipulated, the annotators use the prover interface to request a search for a proof indicating that their annotations are substantiated. First, the system uses lambda calculus reductions to create logical forms that represent the meanings of T_{simple} and H_{simple} in higher-order logic. At this stage, type errors may be reported due to erroneous parse-trees or annotations. In this case an annotator will fix the errors and re-run the proving step. Second, once all type errors are resolved, the higher-order representations are lowered to first order and Prover9 (McCune, 2010) is executed to search for a proof between the logical expressions of T_{simple} and H_{simple}.[6] The proofs are recorded in order to be included in the corpus release. Figure 4 shows the result of translating T_{simple} and H_{simple} to an input to Prover9.

4. Corpus Preparation

We have so far completed annotating 40 positive entailments based on data from RTE 1-4. The annotators are thoroughly familiar with the data and have extensive experience in recognizing entailments stemming from appositive, restrictive and intersective modification. While compiling a corpus of several hundred entailment pairs, we are also working to extend our model to recognize inferences produced by a wider range of linguistic phenomena. The objective is to minimize the need for simplifying the input utterances so as to make them compatible to the model.

```
formulas(assumptions).
% Pragmatics:
all x0 (((nobel_prize(x0) & in_nobel_prize(Physiology_
Medicine, x0)) & great_nobel_prize_in(Physiology_Medicine,
x0)) ↔ x0=c219).
all x0 ((doctor(x0) & australian_doctor(x0)) ↔ x0=c221).
all x0 ((x0=c221 & x0=Robin_Warren) ↔ x0=c220).
all x0 ((bacteria(x0) & ulcer_causing_bacteria(x0)) ↔
x0=c223).
all x0 ((discovery(x0) & of_discovery(c223, x0)) ↔
x0=c222).

% Semantics:
(received(c219, c220) & for_received(c219, c222, c220)).
all x0 (all x1 (received(x0, x1) → awarded(x0, x1))).

end_of_list.

formulas(goals).
exists x0 (nobel_prize(x0) & awarded(x0, Robin_Warren)).
end_of_list.
```

Figure 4: Input for Theorem Prover

5. Conclusions

This paper proposes a novel concept for an annotation platform buttressing a proof-system designed to substantiate a semantic annotation scheme for inferences stemming from modification phenomena. This method guarantees that the manual annotations constitute a complete description of a given entailment relation and facilitates the creation of a

[4] Stanford CoreNLP version 1.3.4

[5] This heuristic is naive in the sense of not disambiguating verbs, adjectives and other types of terms according to their semantic features. It is meant to provide a starting point for the manual annotation process.

[6] Prover9 version 2009-11A

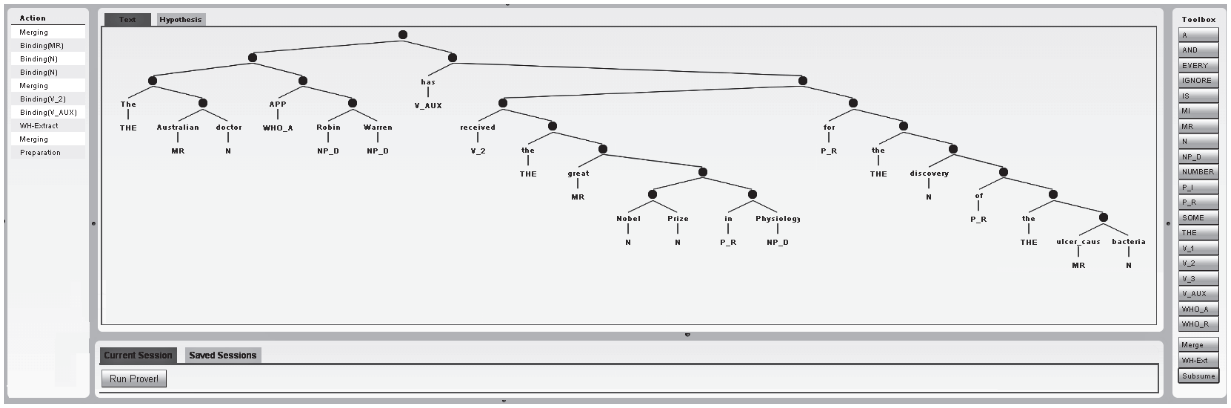

Figure 3: User Interface Panels: Annotation History, Tree-View, Prover Interface and Lexicon Toolbox

gold-standard of such phenomena. A new corpus is currently being developed and will be publicly available for the research community in the foreseeable future.

Acknowledgments

The work of Stavroula Alexandropoulou, Robert Grimm, Sophie Chesney, Pepijn Kokke, Benno Kruit, Kyriaki Neophytou, Antony Nguyen, Assaf Toledo and Yoad Winter was supported by a VICI grant number 277-80-002 by the Netherlands Organisation for Scientific Research (NWO).

6. References

Bar Haim, Roy, Dagan, Ido, Dolan, Bill, Ferro, Lisa, Giampiccolo, Danilo, Magnini, Bernardo, and Szpektor, Idan. (2006). The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Dagan, Ido, Glickman, Oren, and Magnini, Bernardo. (2006). The pascal recognising textual entailment challenge. *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190.

de Marneffe, Marie-Catherine, MacCartney, Bill, and Manning, Christopher D. (2006). Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of the IEEE / ACL 2006 Workshop on Spoken Language Technology*. The Stanford Natural Language Processing Group.

Giampiccolo, Danilo, Dang, Hoa Trang, Magnini, Bernardo, Dagan, Ido, and Cabrio, Elena. (2008). The fourth pascal recognising textual entailment challenge. In *TAC 2008 Proceedings*.

Klein, Dan and Manning, Christopher D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 423–430, Stroudsburg, PA, USA. ACL.

McCune, William. (2010). Prover9 and Mace4. `http://www.cs.unm.edu/~mccune/prover9/`.

Olson, Steven Douglas. (2007). *Ajax on Java*. O'Reilly Media.

Tarski, Alfred. (1944). The semantic conception of truth: and the foundations of semantics. *Philosophy and phenomenological research*, 4(3):341–376.

Toledo, Assaf, Alexandropoulou, Stavroula, Katrenko, Sophia, Klockmann, Heidi, Kokke, Pepijn, and Winter, Yoad. (2013). Semantic Annotation of Textual Entailment. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 240–251, Potsdam, Germany, March. Association for Computational Linguistics.

Toutanova, Kristina, Klein, Dan, Manning, Christopher D., and Singer, Yoram. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA. ACL.