

---

# Towards a Semantic Model for Textual Entailment Annotation

ASSAF TOLEDO,<sup>1</sup> STAVROULA ALEXANDROPOULOU,<sup>1</sup>  
SOPHIE CHESNEY,<sup>2</sup> SOPHIA KATRENKO,<sup>3</sup> HEIDI  
KLOCKMANN,<sup>1</sup> PEPIJN KOKKE,<sup>4</sup> BENNO KRUIT,<sup>5</sup>  
YOAD WINTER<sup>1</sup>

We introduce a new formal semantic model for annotating textual entailments that describes restrictive, intersective, and appositive modification. The model contains a formally defined interpreted lexicon, which specifies the inventory of symbols and the supported semantic operators, and an informally defined annotation scheme that instructs annotators in which way to bind words and constructions from a given pair of premise and hypothesis to the interpreted lexicon. We explore the applicability of the proposed model to the Recognizing Textual Entailment (RTE) 1–4 corpora and describe a first-stage annotation scheme on which we based the manual annotation work. The constructions we annotated were found to occur in 80.65% of the entailments in RTE 1–4 and were annotated with cross-annotator agreement of 68% on average. The annotated parts of the RTE corpora are publicly available for further research.

## 1 Introduction

The *Recognizing Textual Entailment* (RTE) challenges (Dagan et al. 2006) aim to assess a system’s ability to automatically determine whether an entailment relation obtains between a naturally occur-

---

<sup>1</sup>Utrecht University, {a.toledo,s.alexandropoulou,h.e.klockmann,y.winter}@uu.nl

<sup>2</sup>University College London, sophie.chesney.10@ucl.ac.uk

<sup>3</sup>sophia@katrenko.com

<sup>4</sup>Utrecht University, pepijn.kokke@gmail.com

<sup>5</sup>University of Amsterdam, bennokr@gmail.com

ring **text** sentence (T) and a **hypothesis** sentence (H). The RTE corpus (Bar Haim et al. 2006; Giampiccolo et al. 2007, 2008; Ben-tivogli et al. 2009), which is currently the only available resource of textual entailments, marks entailment candidates as valid/invalid.<sup>6</sup> For example:

**Example 1**

T: The head of the Italian opposition, Romano Prodi, was the last president of the EC.

H: Romano Prodi is a former president of the EC.<sup>7</sup>

Entailment: Valid

This categorization contains no indication of the linguistic and informational processes that underlie entailment. In the lack of a gold standard of inferential phenomena, entailment systems can be compared based on their performance, but not on the basis of the linguistic adequacy of their inferential processes. For further remarks on this problem, see Sammons et al. (2010).

The goal of this work is to elucidate some central inferential processes underlying entailments in the RTE corpus. By doing that, we aim to contribute toward creating a benchmark for modeling entailment recognition. We presume that this goal is to be achieved incrementally by modeling increasingly complex semantic phenomena. To this end, we employ a standard model-theoretic approach to entailment that allows combining gold standard annotations with a computational framework. The model contains a formally defined interpreted lexicon, which specifies the inventory of symbols and semantic operators, and an informally defined annotation scheme that instructs annotators how to bind words and constructions from a given T-H pair to entries in the interpreted lexicon. We choose to focus on the semantic phenomena of restrictive, intersective, and appositive modification. This choice is motivated by the predominance of these phenomena in the RTE datasets, the ability to annotate them with high consistency and the possibility of capturing their various syntactic expressions by a limited set of concepts.

For instance, in Example 1 the inference from *The head of the Italian opposition, Romano Prodi*, to *Romano Prodi* is licensed by the semantics of the appositive construction. Lexical phenomena that are not

---

<sup>6</sup>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 *valid*, and the other judgments as *invalid*.

<sup>7</sup>Pair 410 from the test set of RTE 2. *EC* stands for European Commission

modeled but intervene in the analysis of these phenomena, such as the inference *last president*  $\rightarrow$  *former president* in Example 1 are annotated using shallow textual alignment.<sup>8</sup>

In its current stage, this work is only at the beginning of implementing the theoretical semantic model using an annotation platform combined with a theorem prover. In the course of the development of this model, a narrower annotation scheme was adopted. In this scheme, modification phenomena were annotated in all valid entailment pairs from RTE 1–4 without accounting for the way in which the annotated phenomena contributed to the inference being made. This work allowed us to perform data analysis and to further learn about the phenomena of interest as part of the development of the semantic model.

The structure of this paper is as follows. Section 2 reviews some related methods used in Bos et al. (2004) and MacCartney and Manning (2007). In Section 3 we introduce the formal semantic model on which we rely and use it for analyzing some illustrative textual entailments. More RTE data are illustrated in Appendix 1. Section 4 points out a challenge in applying this model to parts of the RTE data and describes our current annotation scheme which aims to address this challenge. We then elaborate on the methods employed in applying this scheme to the datasets of RTE 1–4, and present some quantitative data on the targeted phenomena and inter-annotator agreement. Section 5 concludes.

## 2 Related Work

Bos and Markert (2005) utilize a CCG parser (Bos et al. 2004) to represent the text and hypothesis in discourse representation structures (DRSs, Kamp and Reyle 1993) that encapsulate information on argument structure, polarity, etc. The DRSs of the text and hypothesis are then translated into formulae in first order logic, and a theorem prover is used in order to search whether there is a logical proof from the text formula to the hypothesis formula. The system reached a relatively high precision score of 76% in recognizing the positive cases in RTE 2 but suffered from a very low recall of 5.8%.

MacCartney and Manning (2007)’s system recognizes monotonic relations (or lack thereof) between aligned lexical items in the text and hypothesis and employs a model of compositional semantics to calculate a sentence-level entailment prediction. The recognition of monotonic relations is done using an adapted version of Sanchez Valencia’s Natural Logic (Sánchez Valencia 1991), the alignment between the text and hypothesis is based on a cost function that extends the Levenshtein

---

<sup>8</sup>For more details on this point see Section 3.3.

string-edit algorithm, and the entailment is classified by a decision tree classifier, trained on a small data set of 69 handmade problems. The system was tested on RTE 3 and achieved relatively high precision scores of 76.39% and 68.06% on the positive cases in the development and test sets, respectively. This system also suffers from low recall scores of 26.70% and 31.71%, respectively.

In Bentivogli et al. (2010), a methodology is described for creating specialized entailment data sets by isolating linguistic phenomena relevant for entailment. Pairs of text and hypothesis from the existing corpus were used to generate a set of mono-thematic pairs, each containing one specific phenomenon that takes part in the original entailment. This analysis includes formulation of rules of inferences according to several pre-defined categories, e.g., *argument realization*: “x’s y”  $\rightarrow$  “y of x”. As part of a feasibility study, this methodology was applied to a sample of 90 pairs randomly extracted from RTE 5 and processed by two annotators. Cross-annotator agreement was reported in two scores: complete agreement in pair analysis was found in 64.4% of the pairs, and partial agreement on the phenomena involved in the inferences was calculated by the Dice (1945) coefficient and reached a value of 0.78. This methodology allows a detailed analysis of the entailments in the corpus, but a full analysis of all entailment patterns in the corpus would necessarily involve complex judgments, and this, in turn, would make high cross-annotator agreement very hard to achieve. Moreover, as discussed in Section 4.7, our experience shows that efficient annotation with high cross-annotator agreement is hard to obtain even in more restricted cases which involve less complex judgments.

The model we propose in this work diverges from these approaches in two respects: (a) its first goal is to develop gold standard semantic annotations based on a general formal semantic model; (b) it does not aim to represent phenomena that are not accounted for in this model. For example, consider the following inference, which is based on causal reasoning: *Khan sold nuclear plans*  $\Rightarrow$  *Khan possessed nuclear plans*.<sup>9</sup> Causal reasoning and lexical relations are not part of the semantic phenomena addressed in this paper, and a pattern in the form of *X sold Y*  $\Rightarrow$  *X possessed Y* should be defined ad-hoc by annotators to align the instances of the verbs *sell* and *possess*. This approach allows us to concentrate on the logical aspects of textual entailment, while phenomena involving lexical semantics and world knowledge are handled by a shallow analysis.<sup>10</sup> Our annotation work is to a large extent in line with

<sup>9</sup>This example of causal reasoning is taken from MacCartney and Manning (2007).

<sup>10</sup>Another related work, which approaches inference in natural language as part

the proposal described in Sammons et al. (2010), whose authors appeal to the NLP community to contribute to entailment recognition work by incrementally annotating the phenomena that underlie the inferences in the RTE corpus.

### 3 Theoretical background and RTE examples

To model entailment in natural language, we assume that entailment describes a *preorder* on natural language sentences. Thus, we assume that 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). A computational theory of entailment should describe an approximation of this preorder on natural language sentences. We use a standard model-theoretic extensional semantics, based on the simple *partial order* on the domain of *truth-values*. Each model  $M$  assigns sentences a truth-value in the set  $\{0, 1\}$ . Such a Tarskian theory of entailment is considered adequate if the intuitive entailment preorder on sentences can be described as the pairs of sentences  $T$  and  $H$  whose truth-values  $\llbracket T \rrbracket^M$  and  $\llbracket H \rrbracket^M$  satisfy  $\llbracket T \rrbracket^M \leq \llbracket H \rrbracket^M$  for all models  $M$ . In this section we give the essentials of this model-theoretic approach to entailment that are relevant to the annotated phenomena and illustrate it using a small interpreted lexicon, simplifying the analysis of some representative examples from the RTE.

#### 3.1 Semantic essentials

We adopt a standard semantic framework (Winter 2010) where the types assigned to natural language expressions encode general aspects of their meaning. More specific aspects of an expression’s meaning are encoded by its possible denotations within the typed domain assigned to it. Functional types are standardly defined below.

**Definition 1.** Let  $B$  be a finite non-empty set of basic types. The set of **functional types** over  $B$  is the smallest set  $\mathcal{T}^B$  that satisfies:

- (i)  $B \subseteq \mathcal{T}^B$ ;
- (ii) If  $\tau$  and  $\sigma$  are types in  $\mathcal{T}^B$  then  $(\tau \rightarrow \sigma)$  is also a type in  $\mathcal{T}^B$ .

The ‘ $\rightarrow$ ’ symbol is omitted for perspicuity, and parentheses in types are

---

of a semantic paradigm, is the FraCaS test suite (Cooper et al. 1996). This suite concerns examples that mainly rely on generalized quantification, argument monotonicity, plurality, anaphora resolution, ellipsis, etc. Entailments based on these phenomena are not very common in the RTE data that are analyzed here. Further research is needed in order to integrate data like those in FraCaS into a formal annotation scheme like the one suggested in this paper.

TABLE 1: Types commonly used for natural language expressions

$t$	sentences
$e$	proper names, referential noun phrases
$et$	intransitive verbs and common nouns
$e(et)$	transitive verbs
$(et)t$	quantificational noun phrases
$(et)e$	definite article ( <i>the</i> )
$(et)((et)t)$	determiners ( <i>some, every</i> )
$\tau\tau$	modifiers (adjectives, adverbs, prepositional phrases, negation, relative clauses)
$\tau(\tau\tau)$	coordinators (conjunction, disjunction, restrictive relative pronouns)

erased whenever this does not lead to ambiguity. Thus, the functional type  $(\tau \rightarrow \sigma)$  is denoted ' $\tau\sigma$ '. For our purposes we assume  $\mathbf{B} = \{e, t\}$ , where  $e$  is the type of *entities* and  $t$  is the type of *truth-values*. Table 1 gives some examples for types that are commonly used for various natural language expressions, as will be illustrated in the interpreted lexicon in Section 3.2. The type scheme  $\tau\tau$  for modifiers and the type scheme  $\tau(\tau\tau)$  for coordinators are used for different expressions, with different types instantiating  $\tau \in \mathcal{T}^{\{e,t\}}$ .

For each type  $\tau$  in  $\mathcal{T}^{\mathbf{B}}$  we inductively define a corresponding *domain*  $D_\tau$ . For basic types in  $\mathbf{B}$  the corresponding domains are assigned by assumption. For each non-basic functional type  $\tau\sigma \in \mathcal{T}^{\mathbf{B}}$  we define the corresponding domain by:

$$D_{\tau\sigma} = D_\sigma^{D_\tau} = \text{the set of functions from } D_\tau \text{ to } D_\sigma$$

The domain  $D_t$  of truth-values is assumed to be constant: the set  $\{0, 1\}$  with the natural partial order  $\leq$ . This partial order allows us to semantically capture the preorder relation induced by entailment on natural language sentences. Given a non-empty domain of entities  $D_e = E$ , we refer to the collection of domains  $\mathcal{F}^E = \{D_\tau \mid \tau \in \mathcal{T}^{\{e,t\}}\}$  as the *frame over E*. Thus, the frame over  $E$  contains the respective domain  $E$  of entities, the domain of truth-values, and all the one-place functions that are derived from these two sets. For instance, the following sets are all subsets of  $\mathcal{F}^E$ , where we abbreviate  $\mathbf{2} = \{0, 1\}$ . The type of members in each set is indicated after each set in brackets:

$$\begin{aligned} &E(e), E^E(ee), \mathbf{2}(t), \mathbf{2}^{\mathbf{2}}(tt), E^{\mathbf{2}}(te), \mathbf{2}^E(et), \mathbf{2}^{(E^E)}((ee)t), \\ &(\mathbf{2}^E)^E(eet), E^{(\mathbf{2}^{\mathbf{2}})}((tt)e), (E^{\mathbf{2}})^{\mathbf{2}}(tte) \dots \end{aligned}$$

Let a *lexicon*  $\Sigma$  be a set of terminal symbols (=words). We use a *typing function* TYP for assigning a type to each word in  $\Sigma$ . Using this function and a set of entities  $E$  we define a *model*  $M$  over  $\Sigma$ . This is

done by mapping each word in  $\Sigma$  to a denotation of the appropriate type in the frame  $\mathcal{F}^E$ . This mapping is called an *interpretation function* and is defined as follows.

**Definition 2.** An interpretation function  $I$  over a lexicon  $\Sigma$  with a typing function  $\text{TYP}$  and a set  $E$ , is a function from the words in  $\Sigma$  to the set  $\bigcup \mathcal{F}^E$ , which sends every word  $w \in \Sigma$  to an element in the corresponding typed domain:  $I(w) \in D_{\text{TYP}(w)}$ .

For instance: let *see* be a word of type  $e(et)$  in a typed lexicon  $\Sigma$ . Then for any non-empty set  $E$ , an interpretation function over  $\Sigma$  and  $E$  sends the word *see* to a function in  $(\mathbf{2}^E)^E$ : a function  $f$  from entities to functions from entities to truth-values. By standard Currying, this function  $f$  characterizes a binary relation over  $E$ . Informally speaking, this relation describes who sees who in a given model  $M$ .

A model is a pair of a set of entities  $E$  and an interpretation function  $I$ , which is formally defined below.

**Definition 3.** A model over a lexicon  $\Sigma$  with a typing function  $\text{TYP}$  is a pair  $M = \langle E, I \rangle$  where  $E$  is a non-empty set and  $I$  is an interpretation function over the typed lexicon  $\Sigma$  and the set  $E$ .

Models over a lexicon are used to give denotations to complex expressions over this lexicon. Given a language  $L$  over  $\Sigma$  and a model  $M = \langle E, I \rangle$  over  $\Sigma$ , we assign every parsed expression  $exp$  in  $L$  a *denotation*  $\llbracket exp \rrbracket^M$ . Every terminal expression  $w$  in  $\Sigma$  is assigned the denotation  $\llbracket w \rrbracket^M = I(w)$ . For non-terminal expressions, we assume a binary parse. Thus, any non-terminal expression  $exp$  in  $\Sigma^*$  is assumed to be parsed into two sub-expressions  $exp_1$  and  $exp_2$  (not necessarily in this order). Such complex expressions are assigned a type and a denotation in  $M$  by the following restrictions:

1. If  $exp_1$  (or  $exp_2$ ) is a word in  $\Sigma$ , the type of  $exp_1$  (or  $exp_2$ ) is as defined by the typing function  $\text{TYP}$ .
2. The types of  $exp_1$  and  $exp_2$  are  $\tau\sigma$  and  $\tau$ , respectively.

With these restrictions, we define types and denotations of complex expressions by simple function application. Thus, the type of the expression  $exp$  is  $\sigma$ , and its denotation  $\llbracket exp \rrbracket^M$  in  $M$  is the following element in  $D_\sigma$ :

$$\llbracket exp \rrbracket^M = \llbracket exp_1 \rrbracket^M (\llbracket exp_2 \rrbracket^M), \text{ where } \llbracket exp_1 \rrbracket^M \in D_{\tau\sigma} \text{ and } \llbracket exp_2 \rrbracket^M \in D_\tau$$

In order to describe lexical denotations of words, interpretation functions within models are restricted by suitable *ad hoc* assumptions. Such assumptions are standardly expressed in higher order lambda-calculus (see Section 3.2). Models that satisfy these *ad hoc* restrictions on lexical

denotations are referred to as *intended models*.

A lexicon  $\Sigma$  together with a typing function  $\text{TYP}$  and a specification of intended models is referred to as an *interpreted lexicon*. Assume now that  $T$  and  $H$  are parsed text and hypothesis sentences of type  $t$  over an interpreted lexicon  $\Sigma$ . *Logical entailment* between  $T$  and  $H$  is model-theoretically described as the classical Tarskian property below.

**The truth-conditionality criterion:** *Let  $T$  and  $H$  be parsed expressions of type  $t$ , over an interpreted lexicon  $\Sigma$ . We say that the parsed sentence  $T$  logically entails  $H$  if and only if the relation  $\llbracket T \rrbracket^M \leq \llbracket H \rrbracket^M$  holds between the truth-value denotations of  $T$  and  $H$  in all intended models  $M$ .*

### Example 2

Consider a mini-lexicon  $\Sigma = \{Dan, sat, ate, and\}$  with the types  $e, et, et, (et)((et)(et))$  respectively. Consider intended models where the interpretation function  $I$  assigns the words  $Dan, sat$  and  $ate$  arbitrary denotations **dan**, **sit** and **eat**, and where the word  $and$  is assigned the following denotation in any model:

$$I(and) = \text{AND} = \lambda A_{et}.\lambda B_{et}.\lambda x_e.B(x) \wedge A(x)$$

With the natural binary parses, these intended models explain the entailment  $Dan\ sat\ and\ ate \Rightarrow Dan\ ate$ , since for each intended model  $M$ :

$$\begin{aligned} & \llbracket Dan\ [sat\ [and\ ate]] \rrbracket^M \\ &= ((\text{AND}(\mathbf{eat}))(\mathbf{sit}))(\mathbf{dan}) && \text{analysis} \\ &= (((\lambda A_{et}.\lambda B_{et}.\lambda x_e.B(x) \wedge A(x))(\mathbf{eat}))(\mathbf{sit}))(\mathbf{dan}) && \text{def. of AND} \\ &= \mathbf{sit}(\mathbf{dan}) \wedge \mathbf{eat}(\mathbf{dan}) && \text{func. app. to} \\ & && \mathbf{eat, sit and} \\ & && \mathbf{dan} \\ &\leq \mathbf{eat}(\mathbf{dan}) && \text{def. of } \wedge \\ &= \llbracket Dan\ ate \rrbracket^M && \text{analysis} \end{aligned}$$

### 3.2 An interpreted lexicon

In this section we introduce a small interpreted lexicon that illustrates our treatment of items of some major lexical categories (articles, nouns, (in-)transitive verbs, etc.). Our aim in creating interpreted lexicons like this is to bind words and expressions in annotated RTE sentences to denotations that mark semantic phenomena and allow us to explain



inferential processes in the RTE. This approach is illustrated in Section 3.3.

The lexicon is presented in Table 2. For each word we state its literal form, the type assigned to it, and its denotation in intended models. Denotations that are assumed to be arbitrary in intended models are given in boldface. For example, the intransitive use of the verb *eat* is assigned the type *et* and its denotation **eat** is an arbitrary function of this type. By contrast, other lexical items have their denotations restricted by the intended models. For example, the definite article *the* is assigned the type  $(et)e$ . In each intended model this article denotes the *iota* function – a function from one-place predicates of type *et* to entities of type *e*, which presupposes the uniqueness of the entity that the predicate is true of and returns this entity. This function is denoted ‘THE’ or ‘ $\iota$ ’. The functions that we use for defining denotations are specified in Figure 1.

Several items in the lexicon are assigned more than one type and/or more than one denotation. This is required because of (type) ambiguity in natural language. For example, the adjective *Dutch* has a predicative usage (e.g. *Jan is Dutch*), but it can also serve as a modifier of nouns (*Jan is a Dutch man*). When semantic information on words is described in entailment data, such ambiguities should be resolved.

Below we make some more remarks on the denotations assumed for the lexical entries in Table 2.

- As in Example 2, the coordinator *and* when appearing as a predicate conjunction is analyzed as a function AND mapping any two *et* predicates *A* and *B* to the predicate that sends every entity *e* to the truth-value of the conjunction  $A(x) \wedge B(x)$ . The resulting predicate characterizes the intersection of the sets characterized by *A* and *B*.
- The copula *is* and the article *a* in copular sentences (e.g. *Dan is a man*) denote the identity function on *et* predicates. This  $(et)(et)$  function, which is denoted IS or A, maps each *et* predicate to itself. The same analysis of *is* holds for other copular sentences with predicative adjectives (e.g. *Dan is short/Dutch*). The English copula can also express an equality relation between entities (e.g. *Dan is Jan*). In such cases it is analyzed as the equality relation  $IS_{eq}$ , of type  $e(et)$ .
- The word *some* denotes the existential quantifier SOME, as it is used in intransitive sentences such as *some man ate* (transitive sentences like *Jan saw some man* are not treated here). The SOME function gets two predicates of type *et* (e.g. for *man* and *ate*) and

TABLE 2: An Interpreted Lexicon

Word	Type	Denotation	Remarks
Dan	$e$	<b>dan</b>	proper name
Jan	$e$	<b>jan</b>	proper name
Vim	$e$	<b>vim</b>	proper name
Sue	$e$	<b>sue</b>	proper name
man	$et$	<b>man</b>	common noun
boy	$et$	<b>boy</b>	common noun
nun	$et$	<b>nun</b>	common noun
alien	$et$	<b>alien</b>	common noun
girl	$et$	<b>girl</b>	common noun
sat	$et$	<b>sit</b>	intrans. verb
ate	$et$	<b>eat</b>	intrans. verb
saw	$e(et)$	<b>see</b>	trans. verb
praised	$e(et)$	<b>praise</b>	trans. verb
greeted	$e(et)$	<b>greet</b>	trans. verb
and	$(et)((et)(et))$	AND	pred. conj. (coordinator)
is	$(et)(et)$	IS	copula (modifier)
is	$e(et)$	$IS_{eq}$	copula (equality)
a	$(et)(et)$	A	indef. article (modifier)
the	$(et)e$	THE	def. article (iota)
some	$(et)((et)t)$	SOME	indef. determiner
who	$(et)((et)(et))$	$WHO_R$	res. rel. pronoun (coordinator)
who	$(et)(ee)$	$WHO_A$	app. rel. pronoun
Dutch	$et$	<b>dutch</b> <sub><math>et</math></sub>	int. adjective (predicate)
Dutch	$(et)(et)$	$I_m(\mathbf{dutch}_{et})$	int. adjective (modifier)
black	$et$	<b>black</b> <sub><math>et</math></sub>	int. adjective (predicate)
black	$(et)(et)$	$I_m(\mathbf{black}_{et})$	int. adjective (modifier)
short	$et$	$P_r(\mathbf{short}_{(et)(et)})$	res. adjective (predicate)
short	$(et)(et)$	$R_m(\mathbf{short}_{(et)(et)})$	res. adjective (modifier)
slowly	$(et)(et)$	$R_m(\mathbf{slowly}_{(et)(et)})$	res. adverb (modifier)

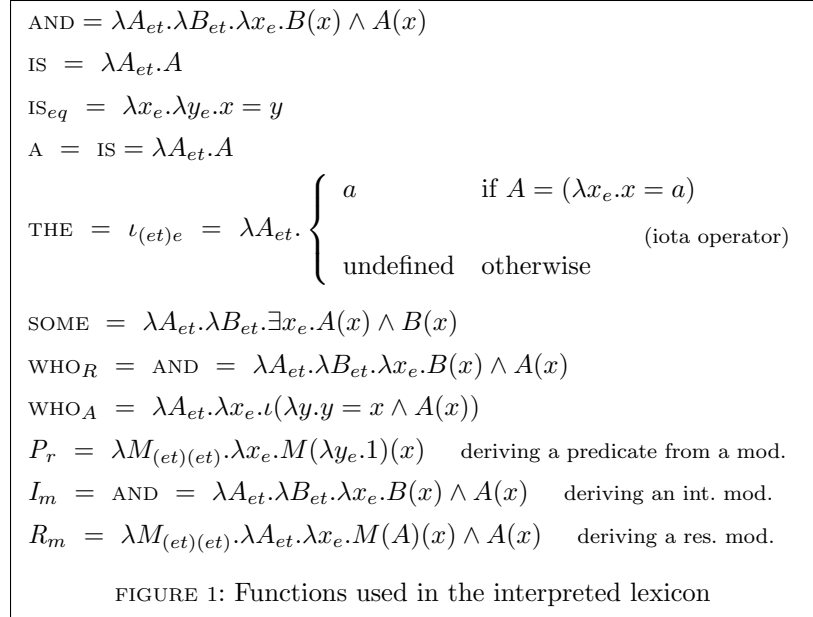
returns 1 iff an entity that satisfies both predicates exists.

- The relative pronoun *who* is an ambiguous form that allows noun modification either by a restrictive relative clause or by an appositive clause. The former is expressed in sentences such as [*the [alien who is a nun]*] *sat*. In this case the pronoun *who* creates a complex predicate, *alien who is a nun* from the predicate constituents *alien* and *is a nun*. The function  $\text{WHO}_R$  that creates this complex predicate for restrictive clauses is the same as the conjunction function  $\text{AND}$ . The appositive use of the pronoun appears in sentences such as [[*the alien*], [*who is a nun*]], *sat*. Here the pronoun adds information on a given entity  $x$  by checking whether a given predicate  $A$  holds of it. In the example the entity  $x$  is denoted by *the alien* and the predicate  $A$  by *is a nun*. The resulting entity is  $x$  if  $A$  holds of  $x$ , and undefined otherwise. This appositive usage of the relative pronoun is defined using the function  $\text{WHO}_A$ .
- The adjectives *short* and *Dutch* have both a modifier entry and a predicative entry. When appearing as modifiers, both adjectives restrict the denotation of the noun they attach to: a *short man* is a *man* and a *Dutch man* is also a *man*. Unlike *short*, the adjective *Dutch* is furthermore *intersective*: a *short man* does not necessarily count as *short* but a *Dutch man* is invariably *Dutch*. The denotations of the adjectives are defined accordingly. The predicate denotation of *Dutch* is defined as an arbitrary constant **dutch** of type *et*. The modifier is derived by this arbitrary predicate and a function  $I_m$  identical to  $\text{AND}$ , which requires entities to satisfy both the noun predicate and the predicate denoted by the constant **dutch**. The denotation of *short* is defined based on the function  $R_m$ , which only requires entities to satisfy the modified noun, after the denotation of the predicate is modified by the arbitrary  $(et)(et)$  function **short**. In the case of *short*, the predicative denotation for *short* is defined using the function  $P_r$  as the set of “short things” – the modifier **short** applied to the whole  $D_e$  domain. The adverb *slowly* is defined only as a restrictive modifier, similar to the use of the adjective *short* for modification.

For more on the theoretical foundations of this approach, see Pratt-Hartmann (2003), Pratt-Hartmann and Moss (2009), Moss (2010a,b), and the references therein.

These works develop theoretical frameworks that aim at wider coverage than what the interpreted lexicon above treats. Integrating all the phenomena treated by Pratt-Hartmann and Moss into an annotation

scheme of the RTE requires further research.



### 3.3 Analyzing entailments using the interpreted lexicon

To illustrate our analysis of entailments using the interpreted lexicon, we give some manual analyses of RTE pairs. The analysis is done by binding expressions in the RTE data to structurally equivalent expressions containing items in the interpreted lexicon. This analysis is three-fold:

1. Phenomena Simplification: we simplify the text and hypothesis to exclude inferential phenomena that we do not handle in the scope of this work. For example, the simplification of *Google operates on the web* to *Google is on the web* is based on lexical knowledge, which we do not address here, and therefore it is handled as part of the simplification step in Example 3.<sup>11</sup>
2. Binding to Lexicon: we bind the constructions in the data to parallel constructions in the interpreted lexicon that share the same

<sup>11</sup>In the future, we intend to capture lexical relations of this kind by adding non-logical axioms to the assumptions based on which a theorem prover aims to find a proof between the logical representation of T and H. Common sense inferences (LoBue and Yates 2011) will be addressed in a similar vein.

structure and semantic properties. This step produces a text sentence  $T_{Lexicon}$  and a hypothesis sentence  $H_{Lexicon}$  as new structurally equivalent versions of the simplified text and hypothesis. We assume parse trees which allow the application of the interpreted lexicon.<sup>12</sup>

3. Proof of Entailment: using predicate calculus and lambda calculus reductions, we establish a logical proof between  $T_{Lexicon}$  and  $H_{Lexicon}$ .<sup>13</sup>

### Example 3

- Data:

T: The largest search engine on the web, Google, receives over 200 million queries each day through its various services.

H: Google operates on the web.<sup>14</sup>

1. Phenomena Simplification:

In the text: adding an overt appositive WH pronoun, for better match with the interpreted lexicon:

$T_{Original}$ : The largest search engine on the web, Google, receives...

$T_{Simple}$ : The largest search engine on the web, which is Google, receives...

In the hypothesis: reducing the meaning of ‘X operates on Y’ to ‘X is on Y’:

$H_{Original}$ : Google operates on the web

$H_{Simple}$ : Google is on the web

2. Binding to Lexicon:

Text:

$T_{Simple}$ : [The largest search engine on the web, which is Google], receives...

$T_{Lexicon}$ : [The short Dutch man, who is Jan], saw Dan

Hypothesis:

$H_{Simple}$ : Google [is [on the web]]

$H_{Lexicon}$ : Jan [is Dutch]

Binding Explanation: we bind the restrictive adjective *largest* to *short*, the noun combination *search engine* to *man* and the inter-

<sup>12</sup>See Kundu and Roth (2011) for previous work that may facilitate binding unknown entries to an existing lexicon.

<sup>13</sup>The only higher-order constants in the above lexicon are the *(et)(et)* constants attributed to non-intersective restrictive modifiers. Treating them in predicate calculus theorem provers may require some *ad hoc* assumptions.

<sup>14</sup>Pair 955 from the test set of RTE 4 (Giampiccolo et al. 2008).

sective modifier *on the web* to *Dutch*. The entity *Google* is bound to *Jan* and the VP *receives...* is bound to *saw Dan*.<sup>15,16</sup>

### 3. Proof of Entailment $T_{Lexicon} \rightarrow H_{Lexicon}$ :

Let  $M$  be an intended model,

$$\begin{aligned}
& \llbracket \llbracket \llbracket \textit{The} [\textit{short Dutch man}], [\textit{who} [\textit{is Jan}],] \textit{saw Dan} \rrbracket \rrbracket \rrbracket^M \\
&= (\text{see}(\mathbf{dan}))((\text{WHO}_A(\text{IS}_{eq}(\mathbf{jan}))) (\iota((R_m(\mathbf{short})) \\
&\quad ((I_m(\mathbf{dutch}))(\mathbf{man})))))) \quad \begin{array}{l} \text{analysis} \\ | \\ \vdots \end{array} \\
&= (\text{see}(\mathbf{dan}))((\text{WHO}_A((\lambda x_e. \lambda y_e. x = \\
&\quad y)(\mathbf{jan}))) (\iota((R_m(\mathbf{short})) ((I_m(\mathbf{dutch}))\mathbf{man})))))) \quad \begin{array}{l} \text{def. of} \\ \text{IS}_{eq} \end{array} \\
&= (\text{see}(\mathbf{dan}))((\text{WHO}_A(\lambda y_e. \mathbf{jan} = y)) \\
&\quad (\iota((R_m(\mathbf{short}))((I_m(\mathbf{dutch}))(\mathbf{man})))))) \quad \begin{array}{l} \text{func. app.} \\ \text{to } \mathbf{jan} \end{array} \\
&= (\text{see}(\mathbf{dan}))((\lambda A_{et}. \lambda x_e. \iota(\lambda y. y = \\
&\quad x \wedge A(x)))(\lambda y_e. \mathbf{jan} = y)) \quad \begin{array}{l} \text{def. of} \\ \text{WHO}_A \end{array} \\
&\quad (\iota((R_m(\mathbf{short}))((I_m(\mathbf{dutch}))(\mathbf{man})))))) \\
&= (\text{see}(\mathbf{dan}))((\lambda x_e. \iota(\lambda y. y = x \wedge (\lambda y_e. \mathbf{jan} = y)(x))) \\
&\quad (\iota((R_m(\mathbf{short}))((I_m(\mathbf{dutch}))(\mathbf{man})))))) \quad \begin{array}{l} \text{func. app.} \\ \text{to } \lambda y_e. \\ \mathbf{jan} = y \end{array} \\
&= (\text{see}(\mathbf{dan}))((\lambda x_e. \iota(\lambda y. y = x \wedge \mathbf{jan} = x)) \\
&\quad (\iota((R_m(\mathbf{short}))((I_m(\mathbf{dutch}))(\mathbf{man})))))) \quad \begin{array}{l} \text{func. app.} \\ \text{to } x \end{array} \\
&= (\text{see}(\mathbf{dan}))(\iota(\lambda y. y = \\
&\quad (\iota((R_m(\mathbf{short}))((I_m(\mathbf{dutch}))(\mathbf{man})))))) \wedge \mathbf{jan} = \\
&\quad (\iota((R_m(\mathbf{short}))((I_m(\mathbf{dutch}))(\mathbf{man})))))) \quad \begin{array}{l} \text{func. app.} \\ \text{to } \iota((R_m( \\ \mathbf{short}))(( \\ I_m(\mathbf{dutch} \\ ))(\mathbf{man}))) \end{array}
\end{aligned}$$

The expression:

$$\begin{aligned}
& \iota(\lambda y. y = (\iota((R_m(\mathbf{short}))((I_m(\mathbf{dutch}))(\mathbf{man})))))) \wedge \\
& \quad \mathbf{jan} = (\iota((R_m(\mathbf{short}))((I_m(\mathbf{dutch}))(\mathbf{man}))))))
\end{aligned}$$

<sup>15</sup>The post-nominal intersective modifier *on the web* is bound to a pre-nominal modifier *Dutch* in order to match the vocabulary of the interpreted lexicon, in which the only intersective modifier is *Dutch*.

<sup>16</sup>In this example,  $T_{Simple}$  (consequently from  $T_{Original}$ ) is structurally ambiguous between *The* [*largest* [*search engine on the web*]], *which is Google, receives...* and *The* [*largest search engine*] *on the web*, *which is Google, receives...* We here illustrate the former analysis. The latter analysis can be handled in a similar vein.

is defined and returns an entity denoted by  $z_e$  only if the following holds:

$$\begin{aligned} \lambda y.y &= (\iota((R_m(\mathbf{short}))((I_m(\mathbf{dutch}))(\mathbf{man})))) \wedge \\ \mathbf{jan} &= (\iota((R_m(\mathbf{short}))((I_m(\mathbf{dutch}))(\mathbf{man})))) = \lambda x_e.x = z_e \quad (1) \end{aligned}$$

From (1) it follows that:

$$\begin{aligned} &(\lambda y.y = (\iota((R_m(\mathbf{short}))((I_m(\mathbf{dutch}))(\mathbf{man})))) \wedge \\ \mathbf{jan} &= (\iota((R_m(\mathbf{short}))((I_m(\mathbf{dutch}))(\mathbf{man}))))(z_e) \\ &= z_e = \iota((R_m(\mathbf{short}))((I_m(\mathbf{dutch}))(\mathbf{man}))) \wedge & \text{func. app.} \\ &\quad \mathbf{jan} = \iota((R_m(\mathbf{short}))((I_m(\mathbf{dutch}))(\mathbf{man}))) & \text{to } z_e \\ &\leq \mathbf{jan} = \iota((R_m(\mathbf{short}))((I_m(\mathbf{dutch}))(\mathbf{man}))) & \text{def. of } \wedge \\ &= \mathbf{jan} = \iota((R_m(\mathbf{short}))(((\lambda A_{et}.\lambda B_{et}.\lambda x_e.B(x) \wedge & \text{def. of } I_m \\ &\quad A(x)) (\mathbf{dutch}))(\mathbf{man}))) & \\ &= \mathbf{jan} = \iota((R_m(\mathbf{short}))((\lambda B_{et}.\lambda x_e.B(x) \wedge & \text{func. app.} \\ &\quad \mathbf{dutch}(x)) (\mathbf{man}))) & \text{to } \mathbf{dutch} \\ &= \mathbf{jan} = \iota((R_m(\mathbf{short}))(\lambda x_e.\mathbf{man}(x) \wedge \mathbf{dutch}(x))) & \text{func. app.} \\ & & \text{to } \mathbf{man} \\ &= \mathbf{jan} = \iota(((\lambda M_{(et)(et)}.\lambda A_{et}.\lambda x_e.M(A)(x) \wedge A(x)) & \text{def. of} \\ &\quad (\mathbf{short}))(\lambda x_e.\mathbf{man}(x) \wedge \mathbf{dutch}(x))) & R_m \\ &= \mathbf{jan} = \iota((\lambda A_{et}.\lambda y_e.(\mathbf{short}(A))(y) \wedge A(y)) & \text{func. app.} \\ &\quad (\lambda x_e.\mathbf{man}(x) \wedge \mathbf{dutch}(x))) & \text{to } \mathbf{short} \\ &= \mathbf{jan} = \iota(\lambda y_e.(\mathbf{short}(\lambda x_e.\mathbf{man}(x) \wedge & \text{func. app.} \\ &\quad \mathbf{dutch}(x)))(y) \wedge (\lambda x_e.\mathbf{man}(x) \wedge \mathbf{dutch}(x))(y)) & \text{to } \lambda x_e. \\ & & \mathbf{man}(x) \wedge \\ & & \mathbf{dutch}(x) \\ &= \mathbf{jan} = \iota(\lambda y_e.(\mathbf{short}(\lambda x_e.\mathbf{man}(x) \wedge & \text{func. app.} \\ &\quad \mathbf{dutch}(x)))(y) \wedge (\mathbf{man}(y) \wedge \mathbf{dutch}(y))) & \text{to } y \\ &= \mathbf{jan} = \iota(\lambda y_e.(\mathbf{short}(\lambda x_e.\mathbf{man}(x) \wedge & \text{from def.} \\ &\quad \mathbf{dutch}(x)))(y) \wedge \mathbf{man}(y) \wedge \mathbf{dutch}(y)) & \text{of } \wedge \end{aligned} \quad (2)$$

The expression:

$$\iota(\lambda y_e.(\mathbf{short}(\lambda x_e.\mathbf{man}(x) \wedge \mathbf{dutch}(x)))(y) \wedge \mathbf{man}(y) \wedge \mathbf{dutch}(y))$$

is defined and returns an entity denoted by  $r_e$  only if the following holds:

$$(3) \quad \lambda y_e.(\mathbf{short}(\lambda x_e.\mathbf{man}(x) \wedge \mathbf{dutch}(x)))(y) \wedge \mathbf{man}(y) \wedge \mathbf{dutch}(y) = \lambda x_e.x = r_e$$

From (3) it follows that:

$$\begin{aligned}
 & (\lambda y_e.(\mathbf{short}(\lambda x_e.\mathbf{man}(x) \wedge \mathbf{dutch}(x)))(y) \wedge \mathbf{man}(y) \\
 & \qquad \qquad \qquad \wedge \mathbf{dutch}(y))r_e \\
 & = (\mathbf{short}(\lambda x_e.\mathbf{man}(x) \wedge \mathbf{dutch}(x)))(r_e) \wedge \mathbf{man}(r_e) \wedge \mathbf{dutch}(r_e) \quad \begin{array}{l} | \text{func. app.} \\ | \text{to } r_e \end{array} \\
 & \leq \mathbf{dutch}(r_e) \quad \quad \quad (4) \quad \begin{array}{l} | \text{def. of } \wedge \end{array}
 \end{aligned}$$

From (2) and (3) it follows that:

$$(5) \qquad \qquad \qquad \mathbf{jan} = r_e$$

From (4) and (5) it follows that:

$$\begin{aligned}
 & \mathbf{dutch}(\mathbf{jan}) \\
 & = (\mathbf{IS}(\mathbf{dutch}))(\mathbf{jan}) \quad \begin{array}{l} | \text{def. of IS} \end{array} \\
 & = \llbracket \text{Jan [is Dutch]} \rrbracket^M \quad \begin{array}{l} | \text{analysis} \end{array}
 \end{aligned}$$

A crucial step in this analysis is our assumption that *on the web* is an intersective modifier of *search engine*. This allows the subsumption of *search engine on the web* by *on the web*. In the interpreted lexicon we describe this behavior using the intersective denotation of the modifier *Dutch*. Let us investigate further the implications of this annotation in the following hypothetical example.

#### Example 4

1. Pair 1

T<sub>1</sub>: Jan is a short Dutch man.

H<sub>1</sub>: Jan is a short man.

In this example there is no intuitive entailment: a short Dutch man may be tall for an average man.

2. Pair 2

T<sub>2</sub>: Jan is a black Dutch man.

H<sub>2</sub>: Jan is a black man.

In this example there is intuitively an entailment: a black Dutch man is a man who is both black and Dutch, hence he is a black man.

From a purely textual/syntactic point of view, these two T-H pairs are indistinguishable. The lexical overlap between the text and hypothesis in both pairs is 100%. This does not allow entailment systems to rely on textual measurements for identifying that the pairs need to be classified differently. Such a perfect score of overlap may lead to a false



positive classification in Pair 1 or conversely, to a false negative in Pair 2. Also syntactically, both *short* and *black* serve as adjectives attached to a noun phrase *Dutch man*. There is nothing in this syntactic configuration to suggest that omitting *Dutch* in Pair 1 might result in a different entailment classification than omitting it in Pair 2. However, from a semantic point of view, based on annotations of abstract relations between predicates and their modifiers, we can correctly analyze both the non-validity of the entailment in Pair 1 and the validity of the entailment in Pair 2.

- Analysis of Pair 1

To validate that there is no entailment between a text and a hypothesis requires showing that there is an intended model  $M = \langle E, I \rangle$  in which there is no  $\leq$  relation between their denotations. Let  $M$  be an intended model that satisfies the following:

- $\mathbf{man}_{et}$  characterizes  $\{\mathbf{dan}, \mathbf{jan}, \mathbf{vim}\}$
- $\mathbf{dutch}_{et}$  characterizes  $\{\mathbf{jan}, \mathbf{vim}\}$
- $\mathbf{short}(\mathbf{man})_{et}$  characterizes  $\{\mathbf{dan}\}$
- $\mathbf{short}(\lambda y_e. \mathbf{man}(y) \wedge \mathbf{dutch}(y))_{et}$  characterizes  $\{\mathbf{jan}\}$

Let us assume parse trees as follows:

- Text: *Jan [is [a [short [Dutch man]]]]*
- Hypothesis: *Jan [is [a [short man]]]*

Let  $M$  be an intended model that satisfies the restrictions above.

Consider the denotations of the text and hypothesis in the model  $M$ :

- Text:

$$\llbracket \text{Jan [is [a [short [Dutch man]]]]} \rrbracket^M$$

$$\begin{aligned}
 &= (\text{IS}(\Lambda((R_m(\mathbf{short}))((I_m(\mathbf{dutch}))(\mathbf{man})))))(\mathbf{jan}) && \text{analysis} \\
 &= (\Lambda((R_m(\mathbf{short}))((I_m(\mathbf{dutch}))(\mathbf{man}))))(\mathbf{jan}) && \text{def. of IS} \\
 &= ((R_m(\mathbf{short}))((I_m(\mathbf{dutch}))(\mathbf{man}))))(\mathbf{jan}) && \text{def. of } \Lambda \\
 &= ((R_m(\mathbf{short}))(((\lambda A_{et}. \lambda B_{et}. \lambda x_e. B(x) \wedge A(x)) && \text{def. of } I_m \\
 &\quad (\mathbf{dutch}))(\mathbf{man}))))(\mathbf{jan}) && \\
 &= ((R_m(\mathbf{short}))((\lambda B_{et}. \lambda x_e. B(x) \wedge \mathbf{dutch}(x)) && \text{func. app. to} \\
 &\quad (\mathbf{man}))))(\mathbf{jan}) && \mathbf{dutch} \\
 &= ((R_m(\mathbf{short}))(\lambda x_e. \mathbf{man}(x) \wedge \mathbf{dutch}(x)))(\mathbf{jan}) && \text{func. app. to} \\
 &\quad \mathbf{man} \\
 &= (((\lambda M_{(et)(et)}. \lambda A_{et}. \lambda y_e. M(A)(y) \wedge A(y))(\mathbf{short})) && \text{def. of } R_m \\
 &\quad (\lambda x_e. \mathbf{man}(x) \wedge \mathbf{dutch}(x)))(\mathbf{jan}) &&
 \end{aligned}$$

$$\begin{aligned}
&= ((\lambda A_{et}.\lambda y_e.(\mathbf{short}(A))(y) \wedge A(y))(\lambda x_e.\mathbf{man}(x) \wedge \mathbf{dutch}(x)))(\mathbf{jan}) && \begin{array}{l} \text{func. app. to} \\ \mathbf{short} \end{array} \\
&= ((\lambda y_e.(\mathbf{short}(\lambda x_e.\mathbf{man}(x) \wedge \mathbf{dutch}(x)))(y) \wedge (\lambda x_e.\mathbf{man}(x) \wedge \mathbf{dutch}(x))(y)))(\mathbf{jan}) && \begin{array}{l} \text{func. app. to} \\ \lambda x_e.\mathbf{man}(x) \wedge \\ \mathbf{dutch}(x) \end{array} \\
&= (\lambda y_e.\mathbf{short}(\lambda x_e.\mathbf{man}(x) \wedge \mathbf{dutch}(x)))(y) \wedge (\mathbf{man}(y) \wedge \mathbf{dutch}(y))(\mathbf{jan}) && \begin{array}{l} \text{func. app. to } y \\ \mathbf{man}(y) \wedge \mathbf{dutch}(y) \end{array} \\
&= (\mathbf{short}(\lambda x_e.\mathbf{man}(x) \wedge \mathbf{dutch}(x)))(\mathbf{jan}) \wedge \mathbf{man}(\mathbf{jan}) \wedge \mathbf{dutch}(\mathbf{jan}) && \begin{array}{l} \text{func. app. to } \mathbf{jan} \\ \mathbf{man}(\mathbf{jan}) \wedge \mathbf{dutch}(\mathbf{jan}) \end{array} \\
&= 1 \wedge 1 \wedge 1 && \text{denotations in } M \\
&= 1 && \text{def. of } \wedge
\end{aligned}$$

• Hypothesis:

$$\llbracket \text{Jan } [is [a [short man]]] \rrbracket^M$$

$$\begin{aligned}
&= (\text{IS}(\text{A}((R_m(\mathbf{short}))(\mathbf{man}))))(\mathbf{jan}) && \text{analysis} \\
&= (\text{A}((R_m(\mathbf{short}))(\mathbf{man}))))(\mathbf{jan}) && \text{def. of IS} \\
&= ((R_m(\mathbf{short}))(\mathbf{man}))(\mathbf{jan}) && \text{def. of A} \\
&= (((\lambda M_{(et)(et)}.\lambda A_{et}.\lambda y_e.M(A)(y) \wedge A(y))(\mathbf{short}))(\mathbf{man}))(\mathbf{jan}) && \text{def. of } R_m \\
&= ((\lambda A_{et}.\lambda y_e.(\mathbf{short}(A))(y) \wedge A(y))(\mathbf{man}))(\mathbf{jan}) && \begin{array}{l} \text{func. app. to} \\ \mathbf{short} \end{array} \\
&= (\lambda y_e.(\mathbf{short}(\mathbf{man}))(y) \wedge \mathbf{man}(y))(\mathbf{jan}) && \begin{array}{l} \text{func. app. to} \\ \mathbf{man} \end{array} \\
&= (\mathbf{short}(\mathbf{man}))(\mathbf{jan}) \wedge \mathbf{man}(\mathbf{jan}) && \text{func. app. to } \mathbf{jan} \\
&= 0 \wedge 1 && \text{denotations in } M \\
&= 0 && \text{def. of } \wedge
\end{aligned}$$

Intuitively, *Jan* can be a man who is considered to be short in the population of Dutch men, hence  $(\mathbf{short}(\lambda x_e.\mathbf{man}(x) \wedge \mathbf{dutch}(x)))(\mathbf{jan})$  would return 1, but not in the population of all men, hence  $(\mathbf{short}(\mathbf{man}))(\mathbf{jan})$  would return 0. This is a direct consequence of having *short* denoting a non-intersective modifier: the set denoted by  $\mathbf{short}(\lambda x_e.\mathbf{man}(x) \wedge \mathbf{dutch}(x))$  is not necessarily a subset of  $\mathbf{short}(\mathbf{man})$ .

- Analysis of Pair 2

Let us assume parse trees as follows:

- Text: *Jan [is [a [black [Dutch man]]]]*
- Hypothesis: *Jan [is [a [black man]]]*

A proof of entailment:

Let  $M$  be an intended model,

$\llbracket \text{Jan [is [a [black [Dutch man]]]]} \rrbracket^M$

$$\begin{aligned}
&= (\text{IS}(\text{A}((I_m(\mathbf{black}))((I_m(\mathbf{dutch}))(\mathbf{man})))))(\mathbf{jan}) && \text{analysis} \\
&= (\text{A}((I_m(\mathbf{black}))((I_m(\mathbf{dutch}))(\mathbf{man}))))(\mathbf{jan}) && \text{def. of IS} \\
&= ((I_m(\mathbf{black}))((I_m(\mathbf{dutch}))(\mathbf{man}))))(\mathbf{jan}) && \text{def. of A} \\
&= ((I_m(\mathbf{black}))(((\lambda A_{et}. \lambda B_{et}. \lambda x_e. B(x) \wedge A(x)) && \text{def. of } I_m \\
&\quad (\mathbf{dutch}))(\mathbf{man}))))(\mathbf{jan}) \\
&= ((I_m(\mathbf{black}))((\lambda B_{et}. \lambda x_e. B(x) \wedge \mathbf{dutch}(x)) && \text{func. app. to} \\
&\quad (\mathbf{man}))))(\mathbf{jan}) && \mathbf{dutch} \\
&= ((I_m(\mathbf{black}))(\lambda x_e. \mathbf{man}(x) \wedge \mathbf{dutch}(x)))(\mathbf{jan}) && \text{func. app. to} \\
& && \mathbf{man} \\
&= (((\lambda A_{et}. \lambda B_{et}. \lambda y_e. B(y) \wedge A(y))(\mathbf{black})) && \text{def. of } I_m \\
&\quad (\lambda x_e. \mathbf{man}(x) \wedge \mathbf{dutch}(x)))(\mathbf{jan}) \\
&= (((\lambda B_{et}. \lambda y_e. B(x) \wedge \mathbf{black}(y))) (\lambda x_e. \mathbf{man}(x) \wedge && \text{func. app. to} \\
&\quad \mathbf{dutch}(x)))(\mathbf{jan}) && \mathbf{black} \\
&= (((\lambda y_e. (\lambda x_e. \mathbf{man}(x) \wedge \mathbf{dutch}(x))(y) \wedge && \text{func. app. to} \\
&\quad \mathbf{black}(y))))(\mathbf{jan}) && \lambda x_e. \mathbf{man}(x) \wedge \\
& && \mathbf{dutch}(x) \\
&= (\lambda y_e. (\mathbf{man}(y) \wedge \mathbf{dutch}(y)) \wedge \mathbf{black}(y))(\mathbf{jan}) && \text{func. app. to } y \\
&= (\mathbf{man}(\mathbf{jan}) \wedge \mathbf{dutch}(\mathbf{jan})) \wedge \mathbf{black}(\mathbf{jan}) && \text{func. app. to } \mathbf{jan} \\
&= \mathbf{dutch}(\mathbf{jan}) \wedge (\mathbf{man}(\mathbf{jan}) \wedge \mathbf{black}(\mathbf{jan})) && \text{def. of } \wedge \\
&\leq \mathbf{man}(\mathbf{jan}) \wedge \mathbf{black}(\mathbf{jan}) && \text{def. of } \wedge \\
&= (\lambda y_e. \mathbf{man}(y) \wedge \mathbf{black}(y))(\mathbf{jan}) && \text{beta reduc. } (\mathbf{jan}) \\
&= (\text{A}(\lambda y_e. \mathbf{man}(y) \wedge \mathbf{black}(y)))(\mathbf{jan}) && \text{def. of A} \\
&= (\text{IS}(\text{A}(\lambda y_e. \mathbf{man}(y) \wedge \mathbf{black}(y))))(\mathbf{jan}) && \text{def. of IS} \\
&= (\text{IS}(\text{A}((\lambda B_{et}. \lambda y_e. B(y) \wedge \mathbf{black}(y))(\mathbf{man}))))(\mathbf{jan}) && \text{beta reduc.} \\
& && (\mathbf{man})
\end{aligned}$$

$$\begin{aligned}
&= (\text{IS}(\text{A}(((\lambda A_{et}.\lambda B_{et}.\lambda y_e.B(y) \wedge A(y))(\mathbf{black})) \\
&\quad (\mathbf{man}))))(\mathbf{jan})) && \begin{array}{l} | \text{beta reduc.} \\ | (\mathbf{black}) \end{array} \\
&= (\text{IS}(\text{A}((I_m(\mathbf{black}))(\mathbf{man}))))(\mathbf{jan})) && \begin{array}{l} | \text{def. of } I_m \\ | \end{array} \\
&= \llbracket \text{Jan [is [a [black man]]}] \rrbracket^M \mathbf{W} && \begin{array}{l} | \text{analysis} \\ | \end{array}
\end{aligned}$$

In this case we rely on the intersectivity of *black*, which in conjunction with the intersectivity of *Dutch* licenses the inference that the set characterized by the *et* function  $\llbracket \text{black [Dutch man]} \rrbracket^M$  equals to the set characterized by  $\llbracket \text{Dutch [black man]} \rrbracket^M$ , which is a subset of the set characterized by  $\llbracket \text{black man} \rrbracket^M$ .

To summarize Example 4, based on semantic information that distinguishes between the restrictive modifier *short* and the intersective modifiers *Dutch* and *black*, we are able to correctly draw a distinction between the non-entailment pair  $T_1$ - $H_1$  and the entailment pair  $T_2$ - $H_2$ . Although these simple examples were constructed here for illustrative purposes, the phenomena of intersectivity and restrictiveness that they illustrate are both relevant for analyzing inferential processes in actual RTE examples such as Example 3. See Appendix 1 for more examples from the RTE.

#### 4 Current Annotation Scheme

In the first stages of our attempt to implement the theoretical model described above, we faced a practical problem concerning the binding of expressions in the RTE data to structurally equivalent expressions in the interpreted lexicon: the lack of a user interface that allows annotators to consistently and effectively annotate RTE data. The root of this problem lies in the intricate ways in which the semantic phenomena that we are concerned with are combined with other phenomena or with each other. Simplifying RTE material to an extent that allows binding it to the lexicon as in the above example is often not straightforward. Consider the following example:

##### Example 5

T: *Comdex – once among the world’s largest trade shows, the launching pad for new computer and software products, and a Las Vegas fixture for 20 years – has been canceled for this year.*

H: *Las Vegas hosted the Comdex trade show for 20 years.*<sup>17</sup>

<sup>17</sup>Pair 214 from the development set of RTE 1 (Dagan et al. 2006).

Validating the entailment in this pair requires a lexical alignment between an expression in the text and the word *hosted* in the hypothesis. However, there is no expression in the text to establish this alignment. In the text, the noun *Comdex* is in an appositive relation with three conjoined predications: (i) *once among the world's largest trade shows*; (ii) *the launching pad for new computer and software products*; and (iii) *a Las Vegas fixture for 20 years*. The third element contains a locative restrictive modification in which *Las Vegas* modifies *fixture*. The apposition licenses the inference that *Comdex is a Las Vegas fixture* and serves as a prerequisite for the alignment: *Comdex is a Las Vegas fixture*  $\Rightarrow$  *Las Vegas hosted Comdex* that simplifies the lexical inference. This alignment is also required for validating the modification by the temporal prepositional phrase *for 20 years* which in the text modifies a noun, *fixture*, and in the hypothesis modifies a verb, *host* – apparently two unrelated lexical items. This example illustrates the difficulty in separating lexical inferences from the semantic relations that underlie the constructions they appear in. In this sense, the manual annotation process that we exemplified in Section 3, in which the stage of *Phenomena Simplification* takes place before the semantic machinery applies, is challenging and requires further investigation with RTE data in order to see what part of the RTE can be annotated using this paradigm, and what elements are needed in order to extend its coverage.

Due to this challenge, and in order to enhance our understanding of the phenomena in the RTE corpora, we adopted a narrower annotation scheme that was carried out on RTE 1–4, named SemAnTE 1.0 – *Semantic Annotation of Textual Entailment*.<sup>18</sup> In this annotation work we focused on valid entailments involving restrictive, intersective, and appositive modification that contribute to the recognition of the entailment.<sup>19</sup> In this approach, a construction is annotated if its semantics is required for validating the entailment, but no account is made of the compositional method in which the meaning of the full sentence is obtained. Annotations were marked in 80.65% of the entailments in the RTE 1–4 corpora and reached cross-annotator agreement of 67.96% on average in four consistency checks. The internal structure of the annotated XML files and a use-case of the annotations for evaluating

<sup>18</sup>The annotated files of SemAnTE are publicly available for download from: <http://logiccommonsense.wp.hum.uu.nl/resources/>

<sup>19</sup>Annotators were instructed to construct a full inferential process informally and then to recognize the contribution of the phenomena we aimed to annotate. This method could be applied efficiently only to valid entailments. Invalid entailments marked as *unknown* exhibit an unidentified relation between the text and hypothesis, and pairs marked as *contradictory* rarely center upon the phenomena in question.

an entailment component in the BIUTEE recognizer (Stern and Dagan 2011) are presented in Toledo et al. (2012). See Garoufi (2007) for other relevant work on semantic analysis and annotation of textual entailment done on RTE 2.

#### 4.1 Phenomena Annotated

Our annotations mark inferences by aligning strings in the text and the hypothesis. This is done by pairing each annotation in the text with a corresponding annotation in the hypothesis that marks the output of the inferential process of the phenomenon in question. In the rest of this section we illustrate the phenomena and underline the annotated part in the text with its correspondence in the hypothesis.

#### 4.2 Restrictive modification (RMOD)

T: *A Cuban<sub>Modifier</sub> American<sub>Modifiee</sub> who is accused of espionage pleads innocent.*

H: *American accused of espionage.*

In this case, *Cuban* modifies *American* and restricts the set of Americans to Cuban Americans. This instance of RMOD validates the inference from *Cuban American* to *American* which is required for establishing the entailment. The intersective nature of the process is not exploited in the actual inference, since the hypothesis does not report that the accused person is Cuban. Thus, only the restrictive property of the modifier *Cuban* is here relevant for the validity of the entailment. More syntactic configurations:

- A verb phrase restricted by a prepositional phrase:

T: *The watchdog International Atomic Energy Agency meets in Vienna<sub>Modifiee</sub> on September 19<sub>Modifier</sub>.*

H: *The International Atomic Energy Agency holds a meeting in Vienna.*

- A noun phrase restricted by a prepositional phrase:

T: *U.S. officials have been warning for weeks of possible terror attacks<sub>Modifiee</sub> against U.S. interests<sub>Modifier</sub>.*

H: *The United States has warned a number of times of possible terrorist attacks.*

#### 4.3 Intersective Modification (CONJ)

T: *Nixon was impeached and became the first president ever to resign on August 9th 1974.*

H: *Nixon was the first president ever to resign.*

This conjunction intersects the two verb phrases *was impeached* and *became the first president ever to resign*. The entailment relies on a subsumption of the full construction to the second conjunct. In addition to canonical conjunctive constructions, CONJ appears also in Restrictive Relative Clauses, whereby the relative clause is interpreted intersectively with the noun being modified:

T: *Iran will soon release eight British servicemen detained along with three vessels.*

H: *British servicemen detained.*

#### 4.4 Appositive modification (APP)

- Appositive subsumption (left part):

T: *Mr. Conway, Iamgold's chief executive officer, said the vote would be close.*

H: *Mr. Conway said the vote would be close.*

- Appositive subsumption (right part):

T: *The country's largest private employer, Wal-Mart Stores Inc., is being sued by a number of its female employees who claim they were kept out of jobs in management because they are women.*

H: *Wal-Mart sued for sexual discrimination.*

- Identification of the two parts of the apposition as referring to one another:

T: *The incident in Mogadishu, the Somali capital, came as U.S. forces began the final phase of their promised March 31 pull-out.*

H: *The capital of Somalia is Mogadishu.*

In addition to appositions, APP is annotated in several more syntactic constructions:

- Non-Restrictive Relative Clauses:

T: *A senior coalition official in Iraq said the body, which was found by U.S. military police west of Baghdad, appeared to have been thrown from a vehicle.*

H: *A body has been found by U. S. military police.*

- Title Constructions:

T: *Prime Minister Silvio Berlusconi was elected March 28 with a mandate to reform Italy's business regulations and pull the economy out of recession.*

H: *The Prime Minister is Silvio Berlusconi.*

#### 4.5 Marking Annotations

Given a pair from the RTE in which the entailment relation obtains between the text and hypothesis, the task for the annotators is defined as follows:

1. Read the data and verify the entailment.
2. Describe informally why the entailment holds.
3. Annotate all instances of RMOD, APP and CONJ that play a role in the inferential process.

#### 4.6 Statistics

The annotated corpus is based on the scheme described above, applied to the datasets of RTE 1–4 (Dagan et al. 2006; Bar Haim et al. 2006; Giampiccolo et al. 2007, 2008). The statistics in Table 3 are based on analysis of the annotations done on RTE 1–4 (development and test sets).

#### 4.7 Consistency Checks

We performed four cross-annotator consistency checks on the annotations of SemAnTE 1.0. In each check we picked a number of entailments that both annotators worked on independently and compared the phenomena that they annotated. We reached cross-annotator consistency on 67.96% of the annotations on average, as reported in Table 5. In the remaining 32% of nonidentical annotations, 24.94% of the annotations differed due to ambiguity in the understanding of the sentences by the annotators, to several possible analyses of the inference, or to limited specification in the annotation scheme (see Appendix 2 for examples). These annotations reflect different legitimate interpretations of the data by the annotators. An annotator error was found only in 9.4% of the annotations that were checked.

#### 4.8 Annotation Platform

The annotations were performed using GATE Developer (Cunningham et al. 2011) and recorded above the original RTE XML files. The annotators used the GATE annotation schemes that were defined to correspond to RMOD, APP and CONJ, as shown in Table 4.<sup>20</sup>

The work was performed in two steps: (1) marking the relevant string in the text using one of the GATE annotation schemes that had been

---

<sup>20</sup>The scheme *rel.clause* appears twice in this table because it is used for annotating non-restrictive relative clauses, expressing appositive modification (APP), and also restrictive relative clauses, expressing intersective modification (CONJ). The phenomena APP and CONJ are annotated using several annotation schemes in order to capture the different syntactic expressions that they allow.



TABLE 3: Counters of annotations in RTE 1–4 separated into development and test sets.

$A_{\#}$  indicates the number of annotations,  $P_{\#}$  indicates the number of entailment pairs containing an annotation and  $P_{\%}$  indicates the portion of annotated pairs relative to the total amount of entailment pairs.

(a) RTE 1									
Ann.	Dev set			Test set					
	$A_{\#}$	$P_{\#}$	$P_{\%}$	$A_{\#}$	$P_{\#}$	$P_{\%}$	$A_{\#}$	$P_{\#}$	$P_{\%}$
APP	97	87	31	161	134	34			
CONJ	90	79	28	126	112	28			
RMOD	180	124	44	243	167	42			
Any	367	210	74	530	297	74			

(b) RTE 2									
Ann.	Dev set			Test set					
	$A_{\#}$	$P_{\#}$	$P_{\%}$	$A_{\#}$	$P_{\#}$	$P_{\%}$	$A_{\#}$	$P_{\#}$	$P_{\%}$
APP	179	149	37	155	135	34			
CONJ	141	119	30	161	144	36			
RMOD	314	205	51	394	236	59			
Any	634	318	80	710	350	88			

(c) RTE 3							(d) RTE 4		
Ann.	Dev set			Test set			Test set		
	$A_{\#}$	$P_{\#}$	$P_{\%}$	$A_{\#}$	$P_{\#}$	$P_{\%}$	$A_{\#}$	$P_{\#}$	$P_{\%}$
APP	188	150	38	166	136	34	259	200	40
CONJ	176	138	35	162	134	34	192	164	33
RMOD	300	201	50	307	193	48	429	271	54
Any	664	329	82	635	328	82	880	413	83

TABLE 4: GATE Annotation Schemes

Phenomenon	Annotation Schemes
RMOD	r_modification
APP	apposition, title, rel_clause
CONJ	conjunction, rel_clause

TABLE 5: Results of Four Consistency Checks.

Each check examined 50-70 annotated pairs from RTE 1–4. In these four checks 66%, 74.11%, 66.67% and 64.66% of the annotations were identical, respectively. On average, 67.96% of the annotations we checked were identical. The rubric *Incorrect Ann.* presents cases of annotations done with an incorrect scheme or with an incorrect scope. *Ambig.-Struct.* are cases of structural or modifier-attachment ambiguity in the text that led to divergent annotations. *Ambig.-Infer.* are cases of divergent annotations stemming from several possible analyses of the inference. *Ambig.-Scheme* refers to instances of divergent annotations due to unclarity or limited specification in the annotation scheme. The last two measures are reported only for the second, third and fourth checks. See Appendix 2 for examples.

Measure	RTE 2	RTE 1+2	RTE 3	RTE 4
Data Source(s)	Dev set	Test sets	Dev+Test sets	Test set
Entailment Pairs	50	70	70	70
Total Ann.	93	112	99	133
Identical Ann.	62	83	66	86
Missing Ann.	2	7	7	10
Incorrect Ann.	10	1	2	2
Ambig.-Struct.	9	16	20	15
Ambig.-Infer.	N/A	8	13	12
Ambig.-Scheme	N/A	0	9	7
Consistency (%)	66.67	74.11	66.67	64.66

defined for the purpose (e.g. *apposition*), and (2) marking a string in the hypothesis that corresponds to the output of the inferential process. The annotation in the hypothesis is done using a dedicated *reference\_to* scheme.

#### 4.9 Connection to the interpreted lexicon approach

Consider the following pair from RTE 2:

##### Example 6

T: *The anti-terrorist court found two men guilty of murdering Shapour Bakhtiar and his secretary Soroush Katibeh, who were found with their throats cut in August 1991.*

H: *Shapour Bakhtiar died in 1991.*

Several entailment patterns in this example can be explained by appealing to the semantics of APP, CONJ and RMOD, as follows:

APP: The appositive modification in *Shapour Bakhtiar and his secretary Soroush Katibeh, who were found with their throats cut in August 1991* licenses the inference that *Shapour Bakhtiar and his secretary Soroush Katibeh were found with their throats cut in August 1991*.

RMOD: The restrictive modification in *August 1991* licenses a subsumption of this expression to *1991*.

CONJ: The conjunction in *Shapour Bakhtiar and his secretary Soroush Katibeh* licenses a subsumption of this expression to *Shapour Bakhtiar*.

By combining these three patterns, we can infer that *Shapour Bakhtiar was found with his throat cut in 1991*. However, additional world knowledge is required to infer that *found with his throat cut* entails *died*. In our current annotation scheme this inference cannot be handled, since lexical alignment of unmodeled phenomena is not supported. This illustrates the limitations of the current annotation scheme and motivates a more robust approach, as proposed in Section 3.

## 5 Conclusions

We have described an on-going attempt to establish a model for analyzing entailment data as specified in the RTE challenges. The long-term aim of this project is to contribute to a theoretically sound model of entailment recognition. We have presented a model that utilizes standard semantic principles and illustrated the way it accounts for textual entailment from the RTE corpora. The model centers upon an interpreted lexicon that comprises words and operators. These elements are used

to represent a fragment of English to which premises and hypotheses may be bound.

We focus on the annotation of semantic phenomena which are predominant in the RTE corpora and can be annotated with high consistency, but which may have several syntactic expressions and therefore allow us to generalize regarding abstract entailment patterns. Non-modeled phenomena that exist in the data are simplified in a preparatory step but cases in which such phenomena are deeply intertwined with the semantic phenomena that we model pose a challenge for the formalization of an annotation scheme.

At a first stage, we carried out a restricted annotation scheme marking instances of restrictive, intersective, and appositive modification in entailment pairs, with no account for the full inferential process between the premise and the hypothesis. These phenomena were found in 80.65% of the entailments in RTE 1–4 and were marked with cross-annotator agreement of 68% on average.

We are currently developing an annotation platform based on a proof system. This platform allows annotators to immediately receive feedback from the prover on the soundness of their annotation. Preliminary results indicate that the theoretical work reported here is useful as the basis for such a platform. Further research is currently being conducted to check the feasibility of such platforms for large-scale model building and further linguistic annotation.

## Appendix 1 – Further examples from the RTE datasets

In this Appendix we illustrate the analysis described in Section 3.3 on several additional examples from RTE 1–4. For each example we specify the steps of *Phenomena Simplification* and *Binding to Lexicon* which enable us to model the entailment.

### Example 1

- Source: RTE 2 test set, pair 282
- Data:
  - $T_{Original}$ : Senator Hill and Foreign Affairs Minister Alexander Downer will host the 20th annual AUSMIN (Australia-United States ministerial consultations) conference at the Adelaide Town Hall.
  - $H_{Original}$ : Alexander Downer will host a conference.
- Phenomena Simplification:
  - $T_{Simple}$ : Hill, who is a Senator, and Alexander Downer, who is the Foreign Affairs Minister, will host the 20th annual con-

ference, AUSMIN, which is Australia-United States ministerial consultations, at the Adelaide Town Hall.

$H_{Simple}$ : Alexander Downer will host a conference.

- Binding to Lexicon:<sup>21</sup>

$T_{Lexicon}$ : Jan, who is a man, and Dan, who is the alien, greeted the tall girl, who is Sue, who is a nun, slowly.

$H_{Lexicon}$ : Dan greeted a boy.

- Binding Explanation: we bind the entities *Hill* and *Alexander Downer* to *Jan* and *Dan* respectively. The nouns *Senator* and *Foreign Affairs Minister* are bound to *man* and *alien* respectively. The expression *will host* is bound to *greeted*, *20th annual* is bound to *tall* and *conference* is bound to *boy*. The appositive *AUSMIN* is bound to *who is Vim* and the relative clause *which is Australia-United States ministerial consultations* is bound to *who is a nun*. The restrictive adverb *at the Adelaide Town Hall* is bound to *slowly*.

## Example 2

- Source: RTE 3 development set, pair 118
- Data:

$T_{Original}$ : According to Nelson Beavers, who is a co-owner of the current company, Carolina Analytical Laboratories, LLC., and has ownership/employment history with Woodson-Tenent and Eurofins, the septic system was installed in the early 1990s.

$H_{Original}$ : Nelson Beavers is one of the owners of Carolina Analytical Laboratories.

- Phenomena Simplification:

$T_{Simple}$ : Nelson Beavers, who is one of the owners of the current company, which is Carolina Analytical Laboratories, LLC. and has ownership/employment history with Woodson-Tenent and Eurofins, said that the septic system was installed in the early 1990s.

$H_{Simple}$ : Nelson Beavers is one of the owners of Carolina Analytical Laboratories.

---

<sup>21</sup>This entailment cannot be validated using the interpreted lexicon described in Section 3.3 because a conjunction of entities is not modeled. We provide this example to illustrate the general method.

- Binding to Lexicon:  
 $T_{Lexicon}$ : Jan, who greeted the man, who is Dan, and saw the nun, praised Vim.  
 $H_{Lexicon}$ : Jan greeted Dan.
- Binding Explanation: we bind the entities *Nelson Beavers* and *Carolina Analytical Laboratories, LLC.* to *Jan* and *Dan* respectively. The noun *current company* is bound to *man*. The predicates *is one of the owners of* and *has ownership/employment history with* are bound to *greeted* and *saw*. The conjunction of entities *Woodson-Tenent* and *Eurofins* is bound to the entity *the nun*. The verb phrase *said that septic system was installed in the early 1990s* is bound to the verb phrase *praised Vim*.

### Example 3

- Source: RTE 4 test set, pair 928
- Data:  
 $T_{Original}$ : Five prisoners were beheaded and got their heads exhibited by the rioters during a violent riot in the “Zwinglio Ferreira” Prison, located in Presidente Vencesla, Brazil.  
 $H_{Original}$ : Five people were killed in a Brazilian prison.
- Phenomena Simplification:  
 $T_{Simple}$ : Five people were killed in the prison, which is in Presidente Vencesla, which is Brazilian.  
 $H_{Simple}$ : Five people were killed in a Brazilian prison.
- Binding to Lexicon:  
 $T_{Lexicon}$ : Jan saw the nun, who is a man, who is Dutch.  
 $H_{Lexicon}$ : Jan saw a Dutch nun.
- Binding Explanation: we bind the noun *Five people* to *Jan*, the verb phrase *were killed in* to *saw* and the noun *prison* to *man*. The predicate *in Presidente Vencesla* is bound to *a man* and the adjective *Brazilian* is bound to *Dutch*.

## Appendix 2 – Legitimate Nonidentical Annotations in Consistency Checks

In this Appendix we demonstrate the categories *Ambig.-Struct.*, *Ambig.-Infer.* and *Ambig.-Scheme* mentioned in Table 5, respectively. Our goal here is to show how nonidentical annotations stem from different legitimate interpretations of the data by the annotators.

**Example 1**

- Category: *Ambig.-Struct.* – different annotations due to structural ambiguity.
- Source: RTE 3 test set, pair 750
- Data:
  - T: The British government has indicated its readiness to allow Argentine companies to take part in the development of oilfields in the Falkland islands' territorial waters.
  - H: The British government is ready to allow Argentine companies to participate in the development of oilfields.
- Annotator 1: Marked *in the Falkland islands' territorial waters* as a modifier of *development of oilfields*, corresponding to the structure: [[development of oilfields][in the Falkland islands' territorial waters]].
- Annotator 2: Marked *in the Falkland islands' territorial waters* as a modifier of *oilfields*, corresponding to the structure: [development of [oilfields [in the Falkland islands' territorial waters]]].

**Example 2**

- Category: *Ambig.-Infer.* – different annotations due to multiple ways of establishing the inference.
- Source: RTE 2 test set, pair 24
- Data:
  - T: Microsoft Corp., on Thursday, posted higher quarterly earnings as revenue rose 12 percent, but its shares fell after the world's largest software market said current quarter sales would fall below Wall Street expectations.
  - H: Microsoft showed revenue growth.
- Annotator 1: Inferred *showed revenue growth* from *posted higher quarterly earnings* and therefore marked *as revenue rose 12 percent* as a restrictive modifier of *posted higher quarterly earnings*.
- Annotator 2: Inferred *showed revenue growth* from *posted higher quarterly earnings, as revenue rose 12 percent* and therefore did not mark a restrictive modifier in this construction.

**Example 3**

- Category: *Ambig.-Scheme*. – different annotations due to limited specification in the annotation scheme.
- Source: RTE 2 development set, pair 154
- Data:
  - T: Clonaid said, Sunday, that the cloned baby, allegedly born to an American woman, and her family were going to return to the United States Monday, but where they live and further details were not released.
  - H: Clonaid announced that mother and daughter would be returning to the US on Monday.
- Problem Description: The annotation scheme of SemAnTE 1.0 does not specify how to mark modification of a non-continuous modifiee. In this case, *Sunday* modifies the combination of *said* and its complement *that the cloned baby, allegedly born to an American woman* but this annotation cannot be marked because the modifiee is made of two separated constituents in the syntax.
- Annotator 1: Annotated only *said* as the modifiee of an RMOD.
- Annotator 2: Did not mark the modification.

**Acknowledgments**

The work of Stavroula Alexandropoulou, Sophie Chesney, Sophia Karentenko, Heidi Klockmann, Pepijn Kokke, Assaf Toledo and Yoad Winter was supported by a VICI grant number 277-80-002 by the Netherlands Organisation for Scientific Research (NWO). We thank Ido Dagan, Philippe de Groote, Ian Pratt and Asher Stern for discussions. Part of the work described in Section 4 was also reported in Toledo et al. (2012) and in Toledo et al. (2013).

**References**

- Bar Haim, R., I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor. 2006. The Second PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.
- Bentivogli, L., E. Cabrio, I. Dagan, D. Giampiccolo, M. L. Leggio, and B. Magnini. 2010. Building Textual Entailment Specialized Data Sets: A Methodology for Isolating Linguistic Phenomena Relevant to Inference. In *Proceedings of LREC 2010*.
- Bentivogli, L., I. Dagan, H. T. Dang, D. Giampiccolo, and B. Magnini. 2009. The Fifth PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of TAC*, vol. 9, pages 14-24.



- Bos, J., S. Clark, M. Steedman, J. R. Curran, and J. Hockenmaier. 2004. Wide-Coverage Semantic Representations from a CCG Parser. In *Proceedings of the 20th international conference on Computational Linguistics*, pages 12–40.
- Bos, J. and K. Markert. 2005. Recognising Textual Entailment with Logical Inference. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 628–635.
- Cooper, Robin, Dick Crouch, Jan Van Eijck, Chris Fox, Josef Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, Steve Pulman, Ted Briscoe, Holger Maier, and Karsten Konrad. 1996. *Using the Framework*. The Fracas Consortium.
- Cunningham, Hamish, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Niraj Aswani, Ian Roberts, Genevieve Gorrell, Adam Funk, Angus Roberts, Danica Damjanovic, Thomas Heitz, Mark A. Greenwood, Horacio Saggion, Johann Petrak, Yaoyong Li, and Wim Peters. 2011. *Text Processing with GATE (Version 6)*. GATE.
- Dagan, Ido, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL Recognising Textual Entailment Challenge. *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Textual Entailment* 3944:177–190.
- Dice, L. R. 1945. Measures of the Amount of Ecologic Association Between Species. *Ecology* 26(3):297–302.
- Garoufi, Konstantina. 2007. *Towards a Better Understanding of Applied Textual Entailment: Annotation and Evaluation of the RTE-2 Ddataset*. Master's thesis, Saarland University.
- Giampiccolo, D., H. T. Dang, B. Magnini, I. Dagan, and E. Cabrio. 2008. The Fourth PASCAL Recognising Textual Entailment Challenge. In *TAC 2008 Proceedings*.
- Giampiccolo, Danilo, 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, RTE '07*, pages 1–9. Stroudsburg, PA, USA: Association for Computational Linguistics.
- Kamp, H. and U. Reyle. 1993. *From Discourse to Logic: Introduction to Model-Theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*, vol. 42. Kluwer Academic Dordrecht, The Netherlands.
- Kundu, G. and D. Roth. 2011. Adapting Text Instead of the Model : An Open Domain Approach. In *CoNLL*.
- LoBue, Peter and Alexander Yates. 2011. Types of Common-Sense Knowledge Needed for Recognizing Textual Entailment. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2, HLT '11*, pages

- 329–334. Stroudsburg, PA, USA: Association for Computational Linguistics.
- MacCartney, B. and C. D. Manning. 2007. Natural Logic for Textual Inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200.
- Moss, Lawrence S. 2010a. Intersecting Adjectives in Syllogistic Logic. In *Proceedings of the 10th and 11th Biennial conference on The mathematics of language, MOL '07/09*, pages 223–237. Berlin, Heidelberg: Springer-Verlag.
- Moss, Lawrence S. 2010b. Syllogistic Logics with Verbs. *Journal of Logic and Computation, special issue on papers from Order, Algebra and Logics* 20(4):947–967.
- Pratt-Hartmann, Ian. 2003. A Two-Variable Fragment of English. *Journal of Logic, Language and Information* 12(1):13–45.
- Pratt-Hartmann, Ian and Lawrence S. Moss. 2009. Logics for the Relational Syllogistic. *The Review of Symbolic Logic* 2:647–683.
- Sammons, M., V. G. Vydiswaran, and D. Roth. 2010. Ask Not What Textual Entailment Can Do for You... In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1199–1208.
- Sánchez Valencia, Victor. 1991. *Studies on Natural Logic and Categorical Grammar*. Ph.D. thesis, University of Amsterdam.
- Stern, Asher and Ido Dagan. 2011. A Confidence Model for Syntactically-Motivated Entailment Proofs. In *Proceedings of RANLP 2011*.
- Toledo, Assaf, Stavroula Alexandropoulou, Sophia Katrenko, Heidi Klockmann, Pepijn Kokke, and Yoad Winter. 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: Association for Computational Linguistics.
- Toledo, Assaf, Sophia Katrenko, Stavroula Alexandropoulou, Heidi Klockmann, Asher Stern, Ido Dagan, and Yoad Winter. 2012. Semantic Annotation for Textual Entailment Recognition. In *Proceedings of the Eleventh Mexican International Conference on Artificial Intelligence (MICAI)*.
- Winter, Y. 2010. *Elements of Formal Semantics*. Unpublished ms., to appear with Edinburgh University Press.