On-demand Injection of Lexical Knowledge for Recognising Textual Entailment

Pascual Martínez-Gómez[1]
pascual.mg@aist.go.jp

Koji Mineshima[2]
mineshima.koji@ocha.ac.jp

Yusuke Miyao[1,3,4]
yusuke@nii.ac.jp

Daisuke Bekki[1,2,3]
bekki@is.ocha.ac.jp

[1]Artificial Intelligence Research Center, AIST
[2]Ochanomizu University
[3]National Institute of Informatics and JST, PRESTO
[4]The Graduate University for Advanced Studies (SOKENDAI)
Tokyo, Japan

Abstract

We approach the recognition of textual entailment using logical semantic representations and a theorem prover. In this setup, lexical divergences that preserve semantic entailment between the source and target texts need to be explicitly stated. However, recognising subsentential semantic relations is not trivial. We address this problem by monitoring the proof of the theorem and detecting unprovable sub-goals that share predicate arguments with logical premises. If a linguistic relation exists, then an appropriate axiom is constructed on-demand and the theorem proving continues. Experiments show that this approach is effective and precise, producing a system that outperforms other logic-based systems and is competitive with state-of-the-art statistical methods.

1 Introduction

Recognising Textual Entailment (RTE) is a challenging NLP application where the objective is to judge whether a text fragment H logically follows from another text fragment T (Dagan et al., 2013). Advances in RTE have potentially positive implications in other areas such as fact checking, question-answering or information retrieval. Solutions to the RTE problem span a wide array of methods. Some methods are purely statistical (Lai and Hockenmaier, 2014; Zhao et al., 2014), where a classifying function is estimated using lexical or syntactic features. Other methods are purely semantic (Bos et al., 2004), where logical formulas that represent the text fragments are constructed and used in a formal proof system. And yet others are hybrid systems (Beltagy et al., 2013), where a combination of statistical features and logical formulas are used to judge entailment relations.

In this paper, we adopt a strategy based on logics, encouraged by the high-performance that these systems achieve in linguistically challenging datasets (Abzianidze, 2015; Mineshima et al., 2015). An important advantage of these systems (including ours) is that they are unsupervised, thus no training data is necessary and no parameters need to be adjusted.

Under the perspective of these logic-based systems, there are mainly two associated challenges when solving RTE problems. The first challenge is to model the logics of the language with the purpose to represent the semantics of text fragments accurately. To this end, we follow the standard practice in formal semantics where the meaning of sentences is represented using logical formulas. The second challenge is to account for lexical relations between text fragments, typically between words or non-compositional phrases. We dedicate our efforts to the latter challenge, and assume that wide coverage linguistic resources are available to signal potential relations between lexical items in text fragments. The question is then how to make the best use of these linguistic resources to close the lexical gap between source and target text fragments.

Our contribution is a precise mechanism that allows to construct and use linguistic axioms on-demand. This mechanism monitors the progress

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 710–720,
Valencia, Spain, April 3-7, 2017.

of a logical proof, detects unprovable sub-goals, and inserts axioms when necessary if a lexical relation is found in an external linguistic resource. These linguistic axioms encode lexical relations between specific segments of the source and target text fragments, thus accounting for lexical divergences that preserve semantic inclusion. To the best of our knowledge, this is the first attempt to integrate on-demand axiom injection into a purely logical natural deduction proof to recognise textual entailment. In the SICK dataset, our system obtains the highest accuracy among the logic systems, and competitive results with respect to machine learning approaches. We believe that our formulation is general enough to be extended to other semantic systems and to introduce lexical knowledge from ontological resources or statistical classifiers efficiently and effectively.

2 Related Work

Our work on recognising textual entailment is primarily inspired by Bos and Markert (2005), where first-order logic interpretations of sentences are used to prove entailment relations with theorem provers and model builders. These semantic interpretations were composed using Boxer (Bos et al., 2004) from derivations of a Combinatory Categorial Grammar (CCG) (Steedman, 2000) automatically obtained by C&C, a wide-coverage CCG parser (Clark and Curran, 2007). This system was later extended into Nutcracker (Bjerva et al., 2014), where WordNet (Miller, 1995) and relations from Paraphrase Database (PPDB) (Ganitkevitch et al., 2013) are used to introduce external linguistic resources to account for lexical divergences (Pavlick et al., 2015). Pavlick et al. (2015) study the characteristics of linguistic relations that may signal entailment or contradiction at subsentential level. However, they ignore the logical context in which these linguistic relations occur in the entailment problem. Moreover, Nutcracker is not a purely logical system in that it uses a proof-approximation method with model-builders.

By contrast, our system is purely logic-based, in that it solely relies on proof constructions based on natural deduction system to make entailment judgements. In addition, as we will see below, a goal-directed proof construction procedure in our system is naturally combined with on-demand axiom injection, as opposed to simply selecting any two arbitrary phrases from T and H that display any linguistic relation.

Beltagy et al. (2013) also use Boxer for their logical semantic representations but assign distributional similarity scores to any two phrases from T and H on-the-fly. Their approach is different from ours in that they use probabilistic logic as an underlying logic. Furthermore, the method to create relevant axioms in Beltagy et al. (2013) is based on a naïve enumeration, and they ignore the logical clues on when two phrases are candidates to be related, which we argue against.

Abzianidze (2015) presents a purely logic-based RTE system that uses CCG parsers and a natural-logic-based tableuax prover. However, his logical representations are based on a non-standard natural logic, which requires the definition of new inference rules for each logical word (e.g. *every*, *some*, *no*) and for which generic theorem provers are not reusable. Regarding the introduction of linguistic knowledge, the author uses only WordNet. However, during the learning phase, he adds missing knowledge manually (e.g. *note* is a hyponym of *paper*), whereas we restrict our results to those automatically generated.

Perhaps the most similar strategies to ours are those of Tian et al. (2014) and Beltagy et al. (2016), where the authors produce on-the-fly knowledge when the hypothesis H cannot be proved. In the work of Tian et al. (2014), propositions between T and H are aligned using logical clues; then, dependency paths are extracted between these propositions and WordNet or word vectors are used to assess the similarity between paths. However, the expressive power of their underlying representation system in Dependency-based Compositional Semantics (DCS) is rather limited and much weaker than the full first-order logic (Liang et al., 2013). Several extensions have been proposed (Tian et al., 2014; Dong et al., 2014), yet these DCS-based inference systems are a non-standard axiomatic system with many axioms and tend to be ad hoc. Whereas their semantic representations are specific to their logic framework, ours are well-understood, logically transparent representations that are generic to most state-of-the-art theorem provers using first-order logic.

Beltagy et al. (2016) use a Modified Robinson Resolution strategy to align clauses and literals between T and H. These alignments also constrain how the unaligned fragments of T and H may correspond to each other, reducing the

problem to a word or phrasal entailment recognition using a statistical classifier. However, that work only considers one possible set of alignments between T-H fragments, which has a decaying coverage when there is repetitions of content words and meta-predicates (typically occurring in medium and long sentences). Instead, we consider multiple alignments by backtracking the decisions on variable and predicate unifications, which is a more powerful strategy. Beltagy et al. (2016) use Markov Logic Networks (MLNs), which is an elegant framework that combines logics and probabilistic reasoning. However, the construction of their Markov Networks is limited by first-order logic, which may pose problems to represent modality or generalised quantifiers. Instead, our logical representations can also be used in a more expressive, higher-order inference system such as the one in Martínez-Gómez et al. (2016), as it was shown by Mineshima et al. (2015) and Mineshima et al. (2016) in a practical application for RTE.

3 Background

This section provides some basic background on our logic-based approach to Recognising Textual Entailment (RTE). RTE is a task of determining whether or not a given text (T) entails a given hypothesis (H). In logic-based approaches, T and H are mapped onto logical formulas; whether T entails H is then determined by checking whether $T \rightarrow H$ is a theorem in a logical system, possibly with the help of a knowledge base.

To obtain logical formulas for input sentences, we use the framework of Combinatory Categorial Grammar (CCG) (Steedman, 2000), a lexicalized grammar formalism that provides a transparent interface between syntax and semantics. We follow the standard method of building compositional semantics in CCG-based systems (Blackburn and Bos, 2005; Bos, 2008), where each syntactic category is schematically assigned a meaning representation formally specified as a λ-term. By combining the meanings of constituent words that appear in a CCG derivation tree, we can obtain a logical formula that serves as a semantic representation of an input sentence.

For semantic representations, we adopt Neo-Davidsonian Event Semantics (Parsons, 1990; Bos, 2008; Jurafsky and Martin, 2009). For instance, the sentence in (1) is mapped not to a simple formula (2) but to a formula (3) that involves an event variable.

(1) John greets Mary.
(2) $\mathsf{greet}(\mathsf{john}, \mathsf{mary})$
(3) $\exists v(\mathsf{greet}(v) \wedge (\mathsf{Subj}(v) = \mathsf{john}) \wedge (\mathsf{Obj}(v) = \mathsf{mary}))$

The sentence (3) expresses that there is an event of greeting such that its subject is John and its object is Mary. In our Neo-Davidsonian approach, every verb is decomposed into a one-place predicate over events and a set of functional expressions such as $\mathsf{Subj}(v) = \mathsf{john}$, which relates an event to its participant.

VP-modifiers such as adverbs and prepositional phrases are also analysed as event predicates. For instance, (4) and (5) are analysed as having the semantic representations in (6) and (7), respectively.

(4) John greets Mary warmly.
(5) John walks to a station.
(6) $\exists v(\mathsf{greet}(v) \wedge (\mathsf{Subj}(v) = \mathsf{john}) \wedge (\mathsf{Obj}(v) = \mathsf{mary}) \wedge \mathsf{warmly}(v))$
(7) $\exists v(\mathsf{walk}(v) \wedge (\mathsf{Subj}(v) = \mathsf{john}) \wedge \exists x(\mathsf{station}(x) \wedge (\mathsf{Goal}(v) = x)))$

There are several advantages of using event semantic formulas as representations for natural language inferences. First, it logically derives an entailment pattern to drop adverbial modifiers, such as the one from (4) to (1) and the one from (5) to *John walks*. Another advantage over simple representations like (2) is that it provides a uniform way of capturing the lexical relationship between verbs. For instance, the hypernym relation between the transitive verb *greet* and the intransitive verb *move* is represented as a simple axiom $\forall v(\mathsf{greet}(v) \rightarrow \mathsf{move}(v))$. This is possible because both verbs are analysed as one-place predicates over events, rather than as predicates with different arities such as $\mathsf{greet}(x, y)$ and $\mathsf{move}(x)$. All these inferences are derivable using the standard first-order logic. For these reasons, event semantic formulas are suitable for the purpose of performing logical inferences with lexical knowledge in our setting.

4 Methodology

4.1 Preliminaries: proving strategy

We adopt *natural deduction* (Prawitz, 1965) as a proof calculus. Here, a typical proving strategy is to decompose the logical formulas of T into atoms (subformulas with no logical connectives) and add them into a pool P of logical premises,

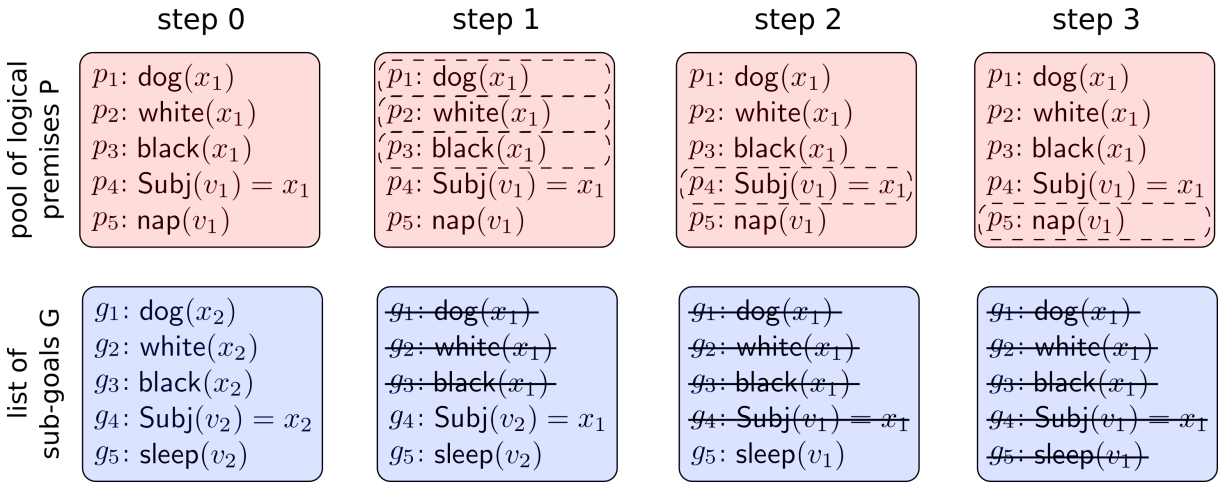

Figure 1: Trace of a proof in natural deduction. In step 0, T and H are decomposed into a pool of logical premises P and a list of sub-goals G. In step 1, g_1, g_2 and g_3 are proved using p_1, p_2 and p_3 and the variable unification $x_2 := x_1$. In step 2, g_4 is proved with p_4 and variable unification $v_2 := v_1$. Finally, g_5 can be proved from p_4 and the external axiom $\forall v.\mathsf{nap}(v) \rightarrow \mathsf{sleep}(v)$, resulting in a proved theorem.

$P = \{p_0(\theta_0), \ldots, p_n(\theta_n)\}$, where p_i are predicates (function names) and θ_i are lists of (possibly structured) arguments of predicates p_i. The logical formula of H is similarly decomposed and its atoms are added either to the pool P or to a list of sub-goals $G = \{p'_0(\theta'_0), \ldots, p'_m(\theta'_m)\}$.

As a running example, consider the T-H pair in (8) and (9), analysed as in (10) and (11):

(8) A black and white dog naps.

(9) A black and white dog sleeps.

(10) $\exists x_1 v_1(\mathsf{dog}(x_1) \wedge \mathsf{white}(x_1) \wedge \mathsf{black}(x_1) \wedge \mathsf{nap}(v_1) \wedge \mathsf{Subj}(v_1) = x_1)$

(11) $\exists x_2 v_2(\mathsf{dog}(x_2) \wedge \mathsf{white}(x_2) \wedge \mathsf{black}(x_2) \wedge \mathsf{sleep}(v_2) \wedge \mathsf{Subj}(v_2) = x_2)$

As we can observe in Figure 1, T would be decomposed into the pool of logical premises

$$P = \{\mathsf{dog}(x_1), \mathsf{white}(x_1), \mathsf{black}(x_1), \mathsf{nap}(v_1), \mathsf{Subj}(v_1) = x_1\}$$

and H into the list of sub-goals

$$G = \{\mathsf{dog}(x_2), \mathsf{white}(x_2), \mathsf{black}(x_2), \mathsf{sleep}(v_2), \mathsf{Subj}(v_2) = x_2\}.$$

In general, existentially quantified formulas whose subformulas are connected only with logical conjunctions (e.g. $\exists\theta.A(\theta) \wedge B(\theta)$) are decomposed into subformulas $A(\theta)$ and $B(\theta)$, and added to P or G if they originate from T and H, respectively. Universally quantified formulas with logical implications (e.g. $\forall\theta.A(\theta) \rightarrow B(\theta)$) are not decomposed if such constructions appear in T; if they appear in H, $B(\theta)$ is added as a sub-goal in G and $A(\theta)$ is added as a logical premise in P. Decomposing higher-order constructions is possible, but we do not treat it here.

The proving then proceeds by selecting a sub-goal $p'_j(\theta'_j)$, searching P for a logical premise $p_i(\theta_i)$ for which p'_j and p_i and their arguments θ'_j and θ_i are equal (or they unify). If such a logical premise is found, then the sub-goal is proved and removed from G. That is the case of the sub-goals g_1 to g_4 in steps 1 and 2 of Figure 1, where predicates match those of p_1 to p_4 and variables unify as $x_2 := x_1$ and $v_2 := v_1$. If all sub-goals are proved, then the theorem is proved and the entailment judgement can be produced.

4.2 Detecting candidate sub-goals

However, there are theorems for which not all sub-goals can be proved. These cases occur when the source text fragment T does not entail the hypothesis H, or when there is a sub-goal for which no premise predicate matches. That is the case of sub-goal g_5 : $\mathsf{sleep}(v_1)$ in Figure 1, which does not match any logical premise p_i. Due to the symbolic nature of logic provers, two different predicates with entailing semantics (e.g. nap and sleep) are considered unrelated, unless stated otherwise. For that reason, such a semantic relation, if it exists, needs to be made explicit in our framework.

In our natural deduction system, this operation is modeled as an *on-line axiom injection*, where candidate sub-goals are detected at proof-

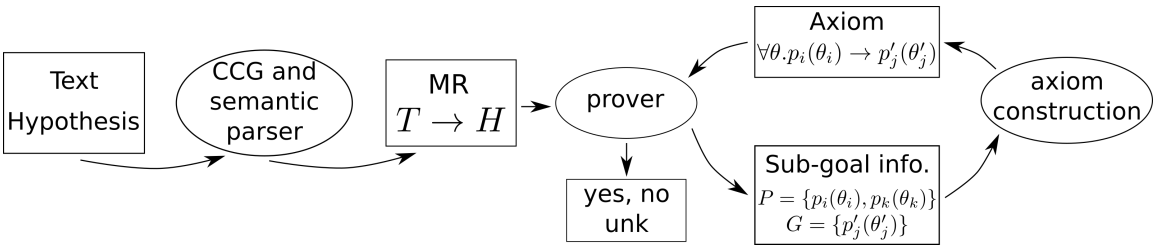

Figure 2: Pipeline for recognising textual entailment. Text and the Hypothesis are syntactically parsed with a CCG parser, and their logical meaning representations (MRs) are composed. A theorem $T \rightarrow H$ is constructed and a prover attempts to test it. If an unprovable sub-goal $p'_j(\theta'_j)$ is found, the axiom construction module attempts to build an axiom $\forall\theta.p_i(\theta_i) \rightarrow p'_j(\theta'_j)$ that is fed back into the theorem.

time, and their semantic relations (if any) with the premises are introduced in the form of axioms.

A sub-goal $p'_j(\theta'_j)$ is detected as a candidate to form an axiom if there is any logical premise $p_i(\theta_i)$ in P such that they share at least one argument, that is, $|\theta'_j \cap \theta_i| > 0$. Instead of requiring the set of arguments θ'_j and θ_i to be equal, we only require them to share at least one argument, to allow sub-goal predicates to underspecify arguments (e.g. drop the object or the subject of the sentence). The set R_j of possible relations between premise predicates p_i and a sub-goal predicate p'_j can then be defined as:

$$R_j = \{p_i \mid p_i(\theta_i) \in T \wedge |\theta'_j \cap \theta_i| > 0\} \quad (1)$$

In the example above, the sub-goal $\mathsf{sleep}(v_1)$ is a candidate sub-goal to form an axiom, and its list of possible relations is $R_{\mathsf{sleep}} = \{\mathsf{nap}\}$.

4.3 On-demand axiom construction

Given a candidate sub-goal $p'_j(\theta'_j)$, R_j is a list of possible predicates that may semantically subsume or exclude the meaning of p'_j. At this point, we only need to classify each $p_i \in R_j$ as subsuming (entailing) p'_j, excluding (contradicting) it, or unrelated. In this work, we choose to use WordNet and VerbOcean (Chklovski and Pantel, 2004) as sources of external linguistic knowledge for their high precision. However, one could use other databases, ontologies or statistical classifiers, but we leave those considerations out of the scope of this paper.

There are two possible types of axioms that can be created: either entailing axioms $\forall\theta.p_i(\theta_i) \rightarrow p'_j(\theta'_j)$, or contradiction axioms $\forall\theta.p_i(\theta_i) \rightarrow \neg p'_j(\theta'_j)$, where $\theta = \theta'_j \cup \theta_i$ is the union of variable names occurring in θ'_j and θ_i. Entailing axioms are created when synonymy (e.g. $\mathsf{house} \rightarrow \mathsf{home}$), hypernymy (e.g. $\mathsf{sea} \rightarrow \mathsf{water}$), adjectival similarity (e.g. $\mathsf{huge} \rightarrow \mathsf{big}$), derivationally related forms (e.g. $\mathsf{accommodating} \rightarrow \mathsf{accommodation}$), or inflection relations (e.g. $\mathsf{wooded} \rightarrow \mathsf{wood}$) are found in WordNet[1]. Contradiction axioms are created solely when antonymy relations (e.g. $\mathsf{big} \rightarrow \neg\mathsf{small}$) are found. Once these axioms are created, they are inserted in the theorem and the proof continues.

Note that in Figure 1, if axioms were created a priori before the proof takes place, an axiom of the form $\forall x.\mathsf{black}(x) \rightarrow \neg\mathsf{white}(x)$ would have been created and a contradiction would be found in step 1 when proving the sub-goal $g_2 : \mathsf{white}(x_1)$. We believe that the frequency of those cases increases with the length of sentences (or paragraphs) and the coverage of the external lexical resources.

Figure 2 shows our pipeline. Our software and Neo-Davidsonian semantic templates are open-sourced and publicly available at `https://github.com/mynlp/ccg2lambda`.

5 Experiments

5.1 Dataset

We use the SemEval-2014 version of the SICK dataset (Marelli et al., 2014), which is a dataset of English single-premise textual entailment problems annotated with three relations: *entailing* (yes), *contradicting* (no) or *unrelated* (unknown). The SICK dataset was originally developed to test approaches of compositional distributional semantics and it includes a variety of lexical, syntactic and semantic phenomena at the sentential level. With respect to FraCaS (Cooper et al., 1994), it contains less linguistically challenging problems but there is a higher need of lexical knowledge,

[1]To maximise coverage, we consider all possible senses for a given predicate (word).

Problem ID	T-H pairs	Entailment
1412	T: *Men are sawing logs .* H: *Men are cutting wood .*	Yes
4114	T: *There is no man eating food .* H: *A man is eating a pizza .*	No
718	T: *A few men in a competition are running outside .* H: *A few men are running competitions outside .*	Unknown

Table 1: Examples of entailment problems from the SICK dataset. Some problems require a mix of logical reasoning and external lexical knowledge.

making it suitable to test our mechanism. With respect to the RTE datasets from the PASCAL RTE challenges, SICK problems are much shorter (and easier to syntactically parse), thus making them affordable for our current semantic parser.

Note that, although the SICK dataset only contains single-premise problems, our method also applies to multi-premise problems out-of-the-box. The dataset contains $4,500$ problems for training, 500 for trial and $4,927$ for testing, with a ratio of yes/no/unk problems of $.29/.15/.56$ in all splits. There are almost $212,000$ running words, an average premise and conclusion length of 10.6 and a vocabulary of $2,409$ words. Typically, there were about 3.6 words in the conclusion that did not appear in the premise, and 3.8 vice versa. Corpus statistics were collected after sentences were tokenized with the Penn Treebank Project tokenizer[2]. Some examples of entailment problems for the SICK dataset are in Table 1.

5.2 Experimental setup

We parsed the tokenized sentences of the premises and hypotheses using the wide-coverage CCG parsers C&C (Clark and Curran, 2007) and EasyCCG (Lewis and Steedman, 2014). CCG derivation trees (parses) were converted into logical semantic representations using ccg2lambda (Martínez-Gómez et al., 2016) and our first-order Neo-Davidsonian event semantics. The validation of our version of semantic templates was carried out exclusively on the trial split of the SICK dataset.

We used Coq (Castéran and Bertot, 2004), an interactive natural deduction (Coquand and Huet, 1988) theorem prover that we run fully automatically with a number of built-in theorem-proving routines called *tactics*, which include first-order logic, arithmetic and equational reasoning. The axiom injection mechanism presented here could also have been implemented as a tactic to achieve a higher proving efficiency. However, this enhancement was left out from this work as it is both technically involved and makes our system bound to this specific prover. Instead, we monitor the proving progress and detect unprovable sub-goals; if our module produces an axiom, then it is introduced in the theorem and the proof is restarted. We call this method **SPSA**, the selector of predicates with shared arguments.

We use two in-house baselines: **No axioms** is our system without axiom injection, where only the logic of the language is used to prove sentence-level entailment relations. **Naïve** is a naïve method where we search for a WordNet linguistic relation between any two words of the premise and conclusion. If such a relation is found, then an axiom is constructed. All axioms found in this way are introduced in the theorem at once, and then the proving is performed. In this naïve method and in SPSA, if two words have more than one WordNet linguistic relation, then we only consider one, in this order: inflections, derivationally related forms, synonyms, antonyms, hypernyms, adjectival similarity and hyponyms. Moreover, although WordNet also contains linguistic relations between phrases, we only consider word-to-word relations. Our plain-logic system, the naïve and the SPSA methods were all timed-out after 100 seconds, at which the entailment judgement "unknown" was produced. When a syntactic parse error occurs, our systems tend to judge the entailment relation as "unknown". To gain robustness and following Abzianidze (2015), we use a multi-parsing strategy (unless stated otherwise), that is, we use both C&C and EasyCCG parsers, and output any of their judgements if they are different

[2] https://www.cis.upenn.edu/~treebank/tokenization.html

from "unknown"[3].

Out of more than 20 participating teams in SemEval 2014, we compare our system to the following representative state-of-the-art systems: **Illinois-LH** (Lai and Hockenmaier, 2014), **ECNU** (Zhao et al., 2014), **UNAL-NLP** (Jiménez et al., 2014), **SemantiKLUE** (Proisl et al., 2014) are systems that build statistical classifiers on shallow features such as word alignments, syntactic structures and distributional similarities. These systems are the top performing systems in SemEval-2014. **The Meaning Factory** (Bjerva et al., 2014) is a hybrid system that combines logic semantic representations derived from CCG trees, with model builders and a statistical classifier, whereas **LangPro** (Abzianidze, 2015) is a purely logic system that composes Lambda Logical Forms of Natural Logic from CCG derivations. **Nutcracker** is a first-order logic system, where the effectiveness of introducing WordNet (and PPDB) using conventional methods is evaluated in (Pavlick et al., 2015).

We also include Markov Logic Networks (MLN) as described by Beltagy et al. (2016), where **MLN** denotes their system with closed-world assumptions and coreferences; **MLN-WN-PPDB** is their system augmented with WordNet and PPDB lexical relations, some handcoded rules, and C&C/EasyCCG multi-parsing; **MLN-eclassif** denotes Beltagy et al. (2016)'s system augmented with a statistical classifier to recognise phrasal entailment relations (hence, we add this system in the list of statistical systems).

As it is common in RTE for SICK, we use precision and recall, where a successful prediction is one where the gold entailment label is either "yes" or "no", and the system correctly predicts it. The accuracy, instead, is computed as a 3-way classification task, where a successful prediction counts on any of the three labels.

5.3 Results

Table 2 shows the results of our experimentation. Our plain first-order logic system **No axioms** has the highest precision 98.90%, but the lowest recall (46.48%). However, its accuracy (76.65%) is well beyond the baseline accuracy (56.69%) based on the majority class.

[3]If the system using C&C parser judges "yes" and the other judges "no", or vice versa, then the final output is "unknown".

System	Prec.	Rec.	Acc.
MLN-eclassif	–	–	85.10
Illinois-LH	81.56	81.87	84.57
ECNU	84.37	74.37	83.64
UNAL-NLP	81.99	76.80	83.05
SemantiKLUE	85.40	69.63	82.32
The Meaning Factory	93.63	60.64	81.60
LangPro Hybrid-800	97.95	58.11	81.35
MLN-WN-PPDB	–	–	80.40
Nutcracker-WN-PPDB	–	–	78.60
Nutcracker-WN	–	–	77.50
Nutcracker	–	–	74.30
MLN	–	–	73.40
Baseline (majority)	–	–	56.69
SPSA-VerbOcean	97.04	63.64	83.13
SPSA	97.07	62.13	82.97
SPSA, only C&C	97.27	58.48	81.44
SPSA, only EasyCCG	97.73	58.71	81.59
Naïve	92.99	59.70	80.98
No axioms	98.90	46.48	76.65

Table 2: Results on the test split of SICK dataset, using precision, recall and accuracy.

The **Naïve** method produced an increase of 4.33% points in accuracy with respect to the pure logic system. As a comparison, Pavlick et al. (2015) reported that a naïve introduction of axioms from WordNet on **Nutcracker** (Bjerva et al., 2014) for SICK dataset leads to an increase of 3.2% points of accuracy (from 74.3% to 77.5%), whereas using WordNet and sophisticated classifiers on the Paraphrase Database (Ganitkevitch et al., 2013) lead to an increase of 4.3% points in accuracy.

When the **SPSA** component substitutes the **Naïve** method, there is a 6.32% increase in the accuracy (from 76.63% to 82.97%), the recall increases by 15.65% and the precision only decreases by 1.83% with respect to the **No axioms** baseline. This system had higher performance than the other two best logic systems **The Meaning Factory** and **LangPro** (82.97% vs. 81.60% and 81.35%), and makes the use of external linguistic knowledge more effective than that in Pavlick et al. (2015), even without the use of a larger paraphrase database such as PPDB. If we add VerbOcean, which is an "unrefined" list of $22,306$ verb relations, the accuracy further improves up to 83.13%, ranking our system on the fourth position among the statistical methods, af-

Prob. ID	T-H pairs	Gold	System	Axioms needed
1412	T: *Men are sawing logs .* H: *Men are cutting wood .*	Yes	Yes	$\forall v.\mathsf{saw}(v) \rightarrow \mathsf{cut}(v)$ $\forall x.\mathsf{log}(x) \rightarrow \mathsf{wood}(x)$
2404	T: *The lady is slicing a tomato .* H: *There is no one cutting a tomato .*	No	No	$\forall v.\mathsf{slice}(v) \rightarrow \mathsf{cut}(v)$
530	T: *A biker is wearing gear which is black .* H: *A biker wearing black is breaking the gears .*	Unk	Yes	
1495	T: *A man is playing a guitar .* H: *A man is strumming a guitar .*	Yes	Unk	$\forall v.\mathsf{play}(v) \rightarrow \mathsf{strum}(v)$
1266	T: *A band is playing on a stage .* H: *A band is playing onstage .*	Yes	Unk	"on a stage" → "onstage"
2166	T: *A woman is sewing with a machine .* H: *A woman is using a machine made for sewing .*	Yes	Unk	"sewing with a machine" → "using a machine made for sewing"
384	T: *A white and tan dog is running through the tall and green grass .* H: *A white and tan dog is running through a field .*	Yes	Unk	"tall and green grass" → "field"

Table 3: Examples of successful and erroneous entailment predictions of our system, collected on the trial split of the SICK dataset.

ter **MLN-eclassif**, **Illinois-LH** and **ECNU**.

When limiting our semantic logical representations to those obtained only from the C&C parser (**SPSA, only C&C**), the recall was reduced by 3.65% and the accuracy by 1.53%, while the precision remained almost equal. Similar results were obtained with the EasyCCG parser. Although no single parser gives clearly a higher performance (in terms of recognising textual entailment), there are clear advantages to using both parsers, which is consistent with findings in Abzianidze (2015).

Regarding the proving time of the SPSA and the naïve methods, there were surprisingly no big differences. The proving time average, median and standard deviation per call to the theorem prover was 10 milliseconds, which was negligible when compared to the python overhead (the main language of our software). The SPSA method did an average of 10.7 calls to the theorem prover per RTE problem, whereas the naïve method did an average of 3.7. Note that these calls include the forward entailment and the contradiction proof attempts, both for C&C and EasyCCG parse trees. If lexical relations were found between the premise and conclusion, the naïve method would only do one more call (for each parser), whereas the SPSA method would do as many calls as axioms are potentially necessary. For that reason, the number of calls of the SPSA method is much larger.

5.4 Positive Examples and Error analysis

Table 3 shows some positive and negative examples of performance of our system on the trial split of the SICK dataset. For the first two examples, our plain logic system (without axioms) produces incorrect entailment judgements ("unknown"), while our system produces the correct label, due to the introduction of two and one axioms in their corresponding theorems, respectively. The first axiom $\forall v.\mathsf{saw}(v) \rightarrow \mathsf{cut}(v)$ states that any event v of sawing is an event of cutting. Note that those predicates only have one argument event variable v, following Neo-Davidsonian event semantics. The second axiom $\forall x.\mathsf{log}(x) \rightarrow \mathsf{wood}(x)$ states that any entity x that is a log is wood.

In the third example, the label of the gold and plain logic system is "unknown". However, our axiom injection system produces the axiom $\forall v.\mathsf{wear}(v) \rightarrow \mathsf{break}(v)$, where the meaning of wear is that of "impairment resulting from long use" (taken from WordNet). These two predicates apply over the object gear, thus sharing the same variable instantiation and producing the error. However, these cases are rare.

In the rest of the examples, our mechanism displays a lack of coverage to create axioms. In the fourth example, play and strum are not direct synonyms, but sister terms (according to WordNet). Many sister terms have an entailment relation, but many others do not (e.g. stand and run). In the rest of the examples, an ideal axiom injection mechanism would need access to string similarity methods (i.e. "on a stage" → "onstage") and to a knowledge base to understand that a machine is something that can be used, or that "tall and green grass" is a "field".

6 Discussion and Future Work

The axiom injection method presented in this paper is more sophisticated and precise than simply assessing the linguistic relation between any two words from T and H and substituting those words conveniently (or equivalently in our framework, to introduce an axiom). Moreover, the naïve method is bound to show gradually lower precision when the size of sentences or the coverage of lexical resources increases, since there are more chances to obtain out-of-context lexical relations. Our axiom injection methods showed larger numbers of calls to the theorem prover, but each call took an average of only 10 milliseconds to complete. The reason for these larger numbers of calls reside in the implementation of the mechanism, since a proof needs to be re-run every time a new axiom is found. A possible enhancement would be to implement our axiom construction and injection as a Coq *tactic* that proves a sub-goal if its predicate has an entailing linguistic relation with any subset of the predicates in the logical premises at a specific stage of a proof.

In order to assess the precision of our SPSA method, we used WordNet and VerbOcean as our databases of external linguistic knowledge, which are databases of high-precision relations. In this setup, we found that our method solves effectively the lack of linguistic knowledge while keeping the precision high. However, other databases such as the Paraphrase Database (Ganitkevitch et al., 2013) or statistical classifiers could further increase the coverage of our method.

On one hand, the SPSA method requires access to the currently active sub-goals (and pool of logical premises) during a proof. Although such information is typically available in logic system, our method might not be directly applicable to systems that rely on statistical classifiers to judge compositional entailment relations. On the other hand, our system is characterised by its very high precision, which is a desirable characteristic when considering system combinations. In such setup, our system could run first, and if no conclusive sentential entailment relation is found, a statistical system could judge the relation, possibly using our logical representations and axioms as features.

Our method cannot be applied yet to larger texts, because CCG derivations accumulate errors when parsing larger sentences, and our logic composition is sensitive to those errors. Thus, making our method more robust against CCG errors is a natural step. One possible solutions is to use N-best CCG trees, collect features from those trees and possibly their semantic logical interpretations, and perform reranking.

7 Conclusion

We have presented a simple and effective method that introduces linguistic axioms on-demand to recognise textual entailments. The strategy is to build logical semantic representation of T and H, monitor the proof of the theorem $T \rightarrow H$, find unprovable sub-goals that share arguments (variable instantiations) with logical predicates, retrieve linguistic knowledge from an external resource, and insert the corresponding axioms on-demand. This system proved more effective and precise than simply enumerating all possible relations between words in T and H.

As it is common in logic systems, our method does not need parameter tuning. Moreover, the semantic representations and axioms are highly interpretable, which makes our system predictable, easy to understand, and easily extensible to use other linguistic resources or classifiers.

Finally, our logics and axiom construction/injection system have a high precision, making it a good candidate either as a standalone system, or as part of larger systems that use our logical semantic interpretations and axioms.

Acknowledgements

This paper is based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO), and is also supported by CREST, JST. We thank the anonymous reviewers for their helpful comments.

References

Lasha Abzianidze. 2015. A tableau prover for natural logic and language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2492–2502, Lisbon, Portugal, September. Association for Computational Linguistics.

Islam Beltagy, Cuong Chau, Gemma Boleda, Dan Garrette, Katrin Erk, and Raymond Mooney. 2013. Montague meets Markov: Deep semantics with probabilistic logical form. pages 11–21, June.

Islam Beltagy, Stephen Roller, Pengxiang Cheng, Katrin Erk, and Raymond J. Mooney. 2016. Representing meaning with a combination of logical and distributional models. *Computational Linguistics*, 42(4):763–808.

Johannes Bjerva, Johan Bos, Rob van der Goot, and Malvina Nissim. 2014. The Meaning Factory: Formal semantics for recognizing textual entailment and determining semantic similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 642–646, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Patrick Blackburn and Johan Bos. 2005. *Representation and Inference for Natural Language: A First Course in Computational Semantics*. CSLI.

Johan Bos and Katja 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. Association for Computational Linguistics.

Johan Bos, Stephen Clark, Mark Steedman, James R Curran, and Julia Hockenmaier. 2004. Wide-coverage semantic representations from a CCG parser. In *Proceedings of the 20th international conference on Computational Linguistics*, pages 104–111. Association for Computational Linguistics.

Johan Bos. 2008. Wide-coverage semantic analysis with Boxer. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, pages 277–286.

Pierre Castéran and Yves Bertot. 2004. *Interactive Theorem Proving and Program Development. Coq'Art: The Calculus of Inductive Constructions.* Springer Verlag.

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

Stephen Clark and James R Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

Robin Cooper, Richard Crouch, Jan van Eijck, Chris Fox, Josef van Genabith, Jan Jaspers, Hans Kamp, Manfred Pinkal, Massimo Poesio, Stephen Pulman, et al. 1994. FraCaS–a framework for computational semantics. *Deliverable*, D6.

Thierry Coquand and Gerard Huet. 1988. The calculus of constructions. *Information and Computation*, 76(2-3):95–120.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. *Recognizing textual entailment: Models and applications*, volume 6. Morgan & Claypool Publishers.

Yubing Dong, Ran Tian, and Yusuke Miyao. 2014. Encoding generalized quantifiers in dependency-based compositional semantics. In *Proceedings of the 28th Pacific Asia Conference on Language, Information, and Computation*, pages 585–594.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, Atlanta, Georgia, June. Association for Computational Linguistics.

Sergio Jiménez, George Dueñas, Julia Baquero, and Alexander Gelbukh. 2014. UNAL-NLP: Combining soft cardinality features for semantic textual similarity, relatedness and entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 732–742, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing*. Prentice-Hall, Inc.

Alice Lai and Julia Hockenmaier. 2014. Illinois-LH: A denotational and distributional approach to semantics. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 329–334, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Mike Lewis and Mark Steedman. 2014. A* CCG parsing with a supertag-factored model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 990–1000, Doha, Qatar, October. Association for Computational Linguistics.

Percy Liang, Michael Jordan, and Dan Klein. 2013. Learning dependency-based compositional semantics. *Computational Linguistics*, 39(2):389–446.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of LREC2014*, pages 216–223.

Pascual Martínez-Gómez, Koji Mineshima, Yusuke Miyao, and Daisuke Bekki. 2016. ccg2lambda: A compositional semantics system. In *Proceedings of ACL-2016 System Demonstrations*, pages 85–90, Berlin, Germany, August. Association for Computational Linguistics.

George A. Miller. 1995. WordNet: A lexical database for English. *Commun. ACM*, 38(11):39–41, November.

Koji Mineshima, Pascual Martínez-Gómez, Yusuke Miyao, and Daisuke Bekki. 2015. Higher-order logical inference with compositional semantics. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2055–2061, Lisbon, Portugal, September. Association for Computational Linguistics.

Koji Mineshima, Ribeka Tanaka, Pascual Martínez-Gómez, Yusuke Miyao, and Daisuke Bekki. 2016. Building compositional semantics and higher-order inference system for a wide-coverage Japanese CCG parser. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2236–2242, Austin, Texas, November. Association for Computational Linguistics.

Terence Parsons. 1990. *Events in the Semantics of English*. MIT Press.

Ellie Pavlick, Johannes Bos, Malvina Nissim, Charley Beller, Benjamin Van Durme, and Chris Callison-Burch. 2015. Adding semantics to data-driven paraphrasing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*, pages 1512–1522. Association for Computational Linguistics.

Dag Prawitz. 1965. *Natural Deduction – A Proof-Theoretical Study*. Almqvist & Wiksell, Stockholm.

Thomas Proisl, Stefan Evert, Paul Greiner, and Besim Kabashi. 2014. SemantiKLUE: Robust semantic similarity at multiple levels using maximum weight matching. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 532–540, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press.

Ran Tian, Yusuke Miyao, and Takuya Matsuzaki. 2014. Logical inference on dependency-based compositional semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 79–89, Baltimore, Maryland, June. Association for Computational Linguistics.

Jiang Zhao, Man Lan, and Tiantian Zhu. 2014. ECNU: Expression- and message-level sentiment orientation classification in twitter using multiple effective features. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 259–264, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.