Extracting higher-order logic formulas from English sentences

Alexandre Rademaker
IBM Research and FGV/EMAp
alexrad@br.ibm.com

Guilherme Lima
IBM Research
guilherme.lima@ibm.com

Renato Cerqueira
IBM Research
rcerq@br.ibm.com

Abstract

We proposed a framework and its implementation as a Python library for converting English utterances into higher-order logic (HOL) formulas. HOL extends first-order logic and provides flexibility for representing natural language semantics. Our library uses a broad-coverage and robust HPSG grammar for English to produce minimal recursive semantics (MRS) structures. These open-source technologies from the DELPH-IN Consortium balance a rigorous linguistic grounding and compositionality with practical aspects for natural language processing applications. Finally, we evaluated our approach over SICK, a popular dataset for text entailment.

1 Introduction

Over the last decades of research on natural language processing (NLP) and computational linguistics (CL), many approaches were proposed for extracting the meaning of linguistic utterances into machine-understandable and unambiguous structures called meaning representations, a task called semantic parsing or semantic analysis (Jurafsky and Martin, 2023). More broadly, the construction and reasoning with meaning representations of natural language expressions are in the context of computational semantics.

This article presents MRS Logic, a library to translate English sentences into logical formulas. MRS Logic is based on methods and tools already extensively studied in the literature and presented in Section 2. Still, it presents some novelty in integrating these resources, our representation language, and how sizeable existing knowledge bases can be easily reused for language understanding.

Consider the ambiguous sentence from Example (1). MRS Logic elucidates all possible interpretations for it, formalizing them in higher-order logic (HOL) expressions. Figure 1 presents two interpretations. Section 3 describes our transformation.

(1) The oil company ensured no chemicals poisoned the river.

From the last paragraph, we can highlight one aspect of our approach: we embrace the ambiguity of natural language. Example (1) has 52 possible interpretations, each representable by a logical expression. Dealing with all possible interpretations and postponing pruning as much as possible may be required by knowledge-intense applications. In many cases, only after linking the linguistic elements to the non-linguistic knowledge of the world can one effectively establish the pragmatics or the speaker's meaning (Quine, 1960; Bender et al., 2015).

Our proposal contrasts with the dominant approach in NLP, where tools shift from explicit symbolic semantic representation to non-compositional and opaque representations such as vector embeddings. Avoiding any strong claim about the requirements for any system that aims at language understanding, we shared some concerns reported in (Mitchell, 2023; Bender et al., 2021) with purely language-model-based tools. Nevertheless, we envision combining large language models (LLM) with symbolic methods in NLP. For instance, extracting relevant common-sense facts from a vast collection of texts.

A well-known problem in the NLP/CL literature is the appropriate metrics for evaluating text understanding and, consequently, the adequacy of the semantic representation formalisms. (Condoravdi et al., 2003) made a case for considering the recognition of text entailment (RTE) between natural language utterances, now broadly considered not a sufficient criterion for language understanding. Still, it remains accepted as a minimal necessary criterion. With that in mind, we evaluate the proposals for semantic representations by measuring their performance on supporting entailment and contradiction detection between pairs of sentences. Section 4 discusses the performance of our system in a balanced subset of a well-known RTE dataset.

$$\exists\, x_{10}, \_oil\_n\_1\ x_{10} \wedge (\exists\, x_{23}, \_river\_n\_of\ x_{23}\ \wedge (\exists\, x_6, (\exists\, e_9, compound\ e_9\ x_6\ x_{10}\ \wedge \_company\_n\_of\ x_6) \wedge (\exists\, e_3, \_ensure\_v\_1\ e_3\ x_6\ (\forall x_{19}, \_chemical\_n\_1\ x_{19} \rightarrow \neg(\exists\, e_{24}, \_poison\_v\_1\ e_{24}\ x_{19}\ x_{23})))))) \quad (1)$$

$$\exists\, x_{10}, \_oil\_n\_1\ x_{10} \wedge (\forall x_{19}, \_chemical\_n\_1\ x_{19} \rightarrow \neg(\exists\, x_{23}, \_river\_n\_of\ x_{23}\ \wedge (\exists\, x_6, (\exists\, e_9, compound\ e8\ x_6\ x_{10}\ \wedge \_company\_n\_of\ x_6) \wedge (\exists\, e_{24}\ e_3, \_ensure\_v\_1\ e_3\ x_6\ (\_poison\_v\_1\ e_{24}\ x_{19}\ x_{23})))))) \quad (2)$$

Figure 1: Two possible logical formulas expressing the possible interpretations for Example (1).

In Section 5, we make some final remarks.

To sum up, our contributions are (1) a framework to produce logical expressions in HOL from English sentences, leveraging ERG grammar and related technologies from the DELPH-IN Consortium; [1] (2) a balanced subset of SICK corpus, sharing our results on the evaluation of our tool on that and some findings.

2 Background

Our library will be described in Section 3, but first, we must describe the technologies we reused and integrated.

The main component of MRS Logic is the English Resource Grammar (ERG) (Flickinger, 2000; Flickinger et al., 2000; Copestake and Flickinger, 2000). The English Resource Grammar is a broad-coverage, linguistically precise, general-purpose computational grammar under continuous development since 1994. It is implemented in the theoretical framework of Head-driven Phrase Structure Grammar (Pollard and Sag, 1994) where both morphosyntactic and semantic properties of English are expressed in a declarative format. Combined with specialized processing tools, it can map running English text to highly normalized representations of meaning called Minimal Recursion Semantics (MRS) (Copestake et al., 2005). ERG is developed as part of the international Deep Linguistic Processing with HPSG Initiative (DELPH-IN). It can be processed by several parsing and realization systems, including the LKB grammar engineering environment (Copestake, 2002), as well as more efficient parsers such as ACE (Crysmann and Packard, 2012).[2]

MRS structures are expressive and have a direct interface with syntax. It can be underspecified in many ways; here, we will describe the underspecification of fine-grained senses and quantifiers' scope. Underspecification allows a single MRS to capture a set of interpretations. Figure 2 shows one among the five possible MRSs for Example (1). It consists of a multiset of relations called elementary predications (EPs). An EP usually corresponds to a single lexeme but can represent grammatical features (e.g., `compound` and `udef_q`, called abstract predicates). Each EP has a label or handle, a predicate symbol, which, in the case of lexical predicates, encodes information about lemma, part-of-speech, and coarse-grained sense distinctions, and a list of numbered arguments: `ARG0`, `ARG1`, etc. The value of an argument can be either a scopal variable (a hole representing the places where alternative labels could fill) or a non-scopal variable (events, states, or entities). The `ARG0` argument has the EP's distinguished variable. This variable denotes an event, state, or referential or abstract entity (e_i or x_i, respectively). Each non-quantifier EP has its unique distinguished variable. Finally, an MRS has a set of handle constraints describing how the EPs' scopal arguments can be nested with EP labels. A constraint $h_i =_q h_j$ denotes equality modulo quantifier insertion. In addition to the indirect linking through handle constraints, EPs are directly linked by sharing the same variable as argument values, capturing the predicate-argument structure of the sentence. Finally, MRS also records properties on variables indicating morpho-syntactic marks of person, number, tense, aspect, etc.

In Figure 2, we see the MRS of the Example (1) where the topmost relation is `_ensure_v_1`, which has the non-empty arguments x_6 and h_{16}. The x_6 is the distinguished variable of the relation `_company_n_of`. A handle constraint equates the sentential variable h_{16} with h_{22}, the label of `_poison_v_1`. The rest of the EPs can be explained similarly. Note that h_5 does not appear in the handle constraints, suggesting that we have more than one possible way to equate this hole with the available labels.

The underspecification of scopes in the MRS of Figure 2 can be represented as the dominance graph (Koller and Thater, 2005) in Figure 3, a directed graph with two kinds of edges: tree edges and dominance edges (in red). Dominance graphs

[1] https://github.com/delph-in/docs/wiki
[2] http://sweaglesw.org/linguistics/ace/

$\langle\, h_1, e_3,$
h_4:_the_q$\langle 0:3\rangle$(ARG0 x_6{PERS *3*, NUM *sg*}, RSTR h_7, BODY h_5),
h_8:compound$\langle 4:15\rangle$(ARG0 e_9{SF *prop*, TENSE *untensed*, MOOD *indicative*, PROG -, PERF -}, ARG1 x_6, ARG2 x_{10}),
h_{11}:udef_q$\langle 4:7\rangle$(ARG0 x_{10}, RSTR h_{13}, BODY h_{12}),
h_{14}:_oil_n_1$\langle 4:7\rangle$(ARG0 x_{10}),
h_8:_company_n_of$\langle 8:15\rangle$(ARG0 x_6, ARG1 i_{15}),
h_2:_ensure_v_1$\langle 16:23\rangle$(ARG0 e_3{SF *prop*, TENSE *past*, MOOD *indicative*, PROG -, PERF -}, ARG1 x_6, ARG2 h_{16}),
h_{17}:_no_q$\langle 24:26\rangle$(ARG0 x_{19}{PERS *3*, NUM *pl*, IND *+*}, RSTR h_{20}, BODY h_{18}),
h_{21}:_chemical_n_1$\langle 27:36\rangle$(ARG0 x_{19}),
h_{22}:_poison_v_1$\langle 37:45\rangle$(ARG0 e_{24}{SF *prop*, TENSE *past*, MOOD *indicative*, PROG -, PERF -}, ARG1 x_{19}, ARG2 x_{23}),
h_{25}:_the_q$\langle 46:49\rangle$(ARG0 x_{23}{PERS *3*, NUM *sg*, IND *+*}, RSTR h_{27}, BODY h_{26}),
h_{28}:_river_n_of$\langle 50:55\rangle$(ARG0 x_{23}, ARG1 i_{29})
$\{\, h_1 =_q h_2,\ h_7 =_q h_8,\ h_{13} =_q h_{14},\ h_{16} =_q h_{22},\ h_{20} =_q h_{21},\ h_{27} =_q h_{28}\,\}\,\rangle$

Figure 2: The first MRS return by ERG for the Example (1).

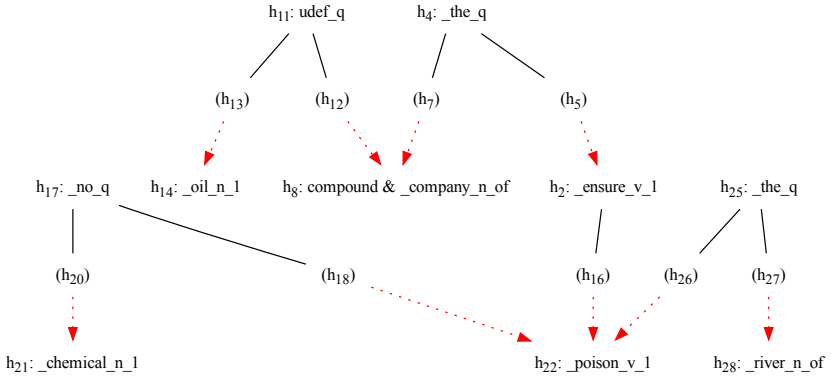

Figure 3: A dominance graph of the MRS from Figure 2.

are used as underspecified descriptions that can be solved to sets of scope trees [3] that later can be realized as formulas in some formal language. Figure 4 shows one of the 32 possible scope trees for the dominance graph from Figure 3. For computing the dominance graph and all possible scope trees for an MRS, we use Utool (Koller and Thater, 2005, 2006, 2010), a GUI and library written in Java.[4].

The scope trees are not directly useful for reasoning, and the CL literature has many proposals for representing NL utterance semantics. One of the most fundamental issues about which logic to use is whether one assumes any structure on the individuals. Other issues are the complexity, decidability, and tools for reasoning in a particular logic. So far, it is reasonable to accept that no existing logic is adequate for all the phenomena of natural language – although we acknowledge different logics individually capture some of the phenomena already studied.

Type theories are widely used in formal theories of the semantics of natural languages (Chatzikyriakidis and Luo, 2020; Ranta, 1994; Winter, 2016).

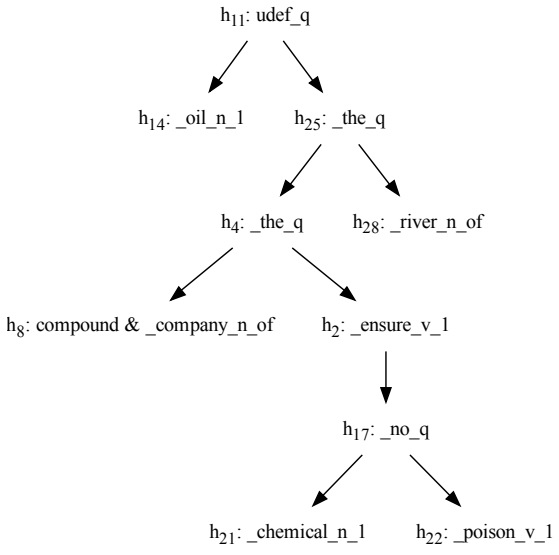

Figure 4: One possible scope tree resolved from the dominance graph from Figure 3.

[3] We are adopting the term suggested by (Emerson, 2020).
[4] https://github.com/coli-saar/utool/

A subset of that, simple type theory, also called higher-order logic (HOL), is a natural extension of first-order logic, which is elegant, highly expressive, and practical (Farmer, 2008). Inspired by modern implementations of simple type theory, such as HOL Light (Harrison, 2009) and Isabelle/HOL (Nipkow et al., 2002), and also by interactive proof assistants based on dependent type theory, such as Lean (Moura and Ullrich, 2021) and Coq (The Coq Development Team, 2021), we implemented the ULKB Logic (Lima et al., 2023). The formulas presented in Figure 1 are HOL formulas encoded in ULKB Logic. ULKB is an open-source framework written in Python for logical reasoning over knowledge graphs. It provides an interactive theorem prover-like environment that can interact with external provers like the E prover (Schulz et al., 2019) and the Z3 SMT solver (de Moura and Björner, 2008).

Finally, consider the possible senses for the word 'company.' ERG only distinguishes senses that are morphosyntactically marked. Since further sense distinctions could never be disambiguated based on grammatical structure alone, the ERG predicate symbol `_company_n_of` intended to be an underspecified representation of all the specific word senses. Wordnet 3.0 (Miller, 1995) contains nine possible nominal senses for this word. We use UKB (Agirre and Soroa, 2009) for Word Sense Disambiguation (WSD), the ERG predicates. It is a collection of programs for performing graph-based and lexical similarity using a pre-existing knowledge base.

3 MRS to Logic

MRS Logic is a library built on top of PyDelphin (Goodman, 2019) and ULKB.[5] It uses PyDelphin to coordinate the call to ERG and iterate over all possible MRS. An MRS is transformed into a scope tree using Utool and finally translated to ULKB formulas. MRS-Logic integrates all technologies described in Section 2. This section describes the translation of scope trees into ULKB formulas, skipping the implementation details of data structures and some design decisions.

At the high level, the translation starts from the topmost node of the scope tree, the handle in the higher position, usually a quantifier. The transformation sketched out in Figure 5 considered the scope tree from Figure 4 as input, and it works recursively.

The node h_{11} is the implicit quantifier udef_q,[6], as all other ERG quantifiers, it is modeled as a generalized (binary) quantifier (Westerståhl, 2019). We interpret this predicate as an existential quantifier in HOL. Nodes h_4 and h_{25} have the same interpretation but are surface predicates.[7] Node h_{17} is another quantifier; our current interpretation is as a universally quantified implication of the restriction to the negation of the body.

Note that variable x_{10} is instantiated, and further transformations of nodes h_{14} and h_{25} will be under the scope of this existential quantifier. Nodes h_{14}, h_{21} are trivial; ERG predicates are transformed into HOL predicates with the same arity. Node h_{28} has one uninstantiated parameter; the lexical entry for _river_n_of in ERG expects one optional complement.[8] Since the parameter was not supplied in the sentence, we decreased the cardinality of the generated HOL predicate. This behavior is configurable in our transformation and may be disabled if needed. The same simplification happens in transforming the predicate _company_n_of in h_8.

Node h_{22} has a verbal predicate with an event variable as its distinguished variable, ARG0 . Event variables are not explicitly quantified in MRS, so we must decide when to introduce them in the HOL formula. The problem is that the existential quantifier for the event should not get a broad scope if negation is involved. Consider the sentence 'No man is walking' and a problematic translation to $\exists\, e2, \forall\, x3, \_man\_n\_1\ x3 \rightarrow \neg\_walk\_v\_1\ e2\ x3$. We don't want to instantiate e_2 to say later that it didn't exist. The correct approach is to instantiate the event variables as close as possible to the predicate with this variable as its distinguished variable.

Node h_2 is where HOL stands out. The verb 'ensure' can be taken as a factive verb (Hazlett, 2010) introducing a presupposition; that is, the HOL predicate gets a HOL formula as an argument, a higher-order construction not permitted in FOL.[9] In this example, this is the only case where $\mathbf{T}$ is applied recursively in a predicate argument.

[5] The code is available at https://github.com/ibm/mrs-logic.

[6] https://github.com/delph-in/docs/wiki/ErgSemantics_ImplicitQuantifiers

[7] https://github.com/delph-in/docs/wiki/ErgSemantics_Basics

[8] Consider 'the river of Colorado.'

[9] We acknowledge that FOL translations for the same phenomena are possible (Bos, 2014).

We haven't yet introduced in the system extra axioms to impose the presupposition reading when needed. Still, the translation presented here would not be affected by such additional axioms once we have a complete understanding of them.[10]

Finally, node h_8 is the only one with more than one EP; the transformation considers all EP with the same label as a coordination of the translations of each EP. We notice the ERG predicate `compound`; it can be considered an underspecified preposition. ERG analyses noun-noun compounds so that `compound` has the same structure as other explicit prepositions, e.g., 'boxes on tables are blue', 'boxes for tables are blue,' and 'table boxes are blue.'

The transformation creates the HOL predicates inline, but it could also pre-declared them as polymorphic predicates, such as $\_oil\_n\_1 : a \rightarrow bool$ where a is a type variable in ULKB.[11]

The translation covers some additional phenomena not illustrated in the example we used. Nevertheless, presenting it in the way we did gives the reader better intuition about its general ideas. We plan to extend our translation to all other ERG abstract predicates that model additional NL phenomena like normalization, disjunctions, conjunctions, etc.

4 SICK Experiment

The SICK dataset includes 9,840 sentence pairs taken from images and video captions. A selection of sentences from each source was used to produce pairs of sentences in 3 steps detailed reported in (Marelli et al., 2014; Bentivogli et al., 2016). The pairs were manually annotated regarding semantic similarity and logical relation: entailment, contradiction, or neutral.[12] The sentence pairs are rich in lexical, syntactic, and semantic phenomena. Still, the entailment test of the sentences was expected to be supported by common sense and grammatical knowledge and not to require encyclopedic knowledge about entities of the world.

Given the sentence's intended simplicity compared to previous datasets for Recognising Textual Entailment (RTE), SICK is excellent for testing our tool. Suppose our translation effectively captures the meaning of the sentences in HOL expressions. If sentences A and B are classified as an entailment, we should be able to prove $\Delta \vdash \mathbf{T}(A) \rightarrow \mathbf{T}(B)$ where Δ is a background theory.[13] If they are classified as a contradiction, we should be able to prove $\Delta \vdash \neg(\mathbf{T}(A) \wedge \mathbf{T}(B))$; otherwise, we consider them as neutral. This is a very simple approach compared to other logical-based RTE reports (Bos, 2014), but our goal here is to have a preliminary test of our transformation, not to improve the results on the SICK leaderboard.[14]

We start pre-parsing all sentences with ERG, asking for at most ten readings for each sentence. Of the 6,077 unique sentences, 3,435 sentences have the maximum number of readings we requested. 2,055 sentences had less than five readings, 564 between 5 and 9 readings, and only 23 sentences were not parsed by ERG, some of them by being ungrammatical (Kalouli et al., 2017a,b). This shows that we have a high degree of ambiguity, even for a collection of relatively simple sentences. Given that, during the main loop of the experiment, when we test each pair for logical entailment, we check at most four combinations of interpretations (HOL formulas) in a breath-first search strategy.

Unfortunately, SICK is very unbalanced regarding the entailment test, as we can see in Table 1, and the corpus contains a lot of repeated sentences. To overcome these limitations, we created a subset of SICK, called SB-SICK (small and balanced SICK), with 330 pairs for each label and no sentence repetition.

#	label
1424	CONTRADICTION
2822	ENTAILMENT
5596	NEUTRAL

Table 1: distribution of SICK sentences for each entailment label.

Table 2 summarizes the results we obtained. In this experiment, we did not use the WSD module, relying on the ERG predicates and its coarse-grained senses. The Δ is a small theory of 24 axioms we added incrementally during the tests, experimenting with the system's adaptabil-

[10] Note that presuppositions are one of many NL phenomena where the deep language processing with ERG, with a curated lexicon, and the kind of semantic analysis we are carrying on here makes the difference. For instance, if we take 'ensure' as a factive verb, we can adequately formalize its meaning. If an entity X ensures Y, Y should be taken as a true statement?

[11] A configuration can also specify a single type for all predicate parameters.

[12] We are only interested in the logical relations.

[13] The $\mathbf{T}(a)$ means the transformation of the NL sentence A into HOL formulas as described in Section 3.

[14] https://paperswithcode.com/dataset/sick

$$\mathbf{T}[h_{11}] = \mathbf{T}[\text{udef\_q ARG0 } x_{10} \text{ RSTR } h_{14} \text{ BODY } h_{25}] = (\exists\, x_{10}, \mathbf{T}[h_{14}] \wedge \mathbf{T}[h_{25}])$$
$$\mathbf{T}[h_{14}] = \mathbf{T}[\text{\_oil\_n\_1 ARG0 } x_{10}] = \_oil\_n\_1(x_{10})$$
$$\mathbf{T}[h_{28}] = \mathbf{T}[\text{\_river\_n\_of ARG0 } x_{23} \text{ ARG1 } i_{29}] = \_river\_n\_of(x_{23})$$
$$\mathbf{T}[h_{21}] = \mathbf{T}[\text{\_chemical\_n\_1 ARG0 } x_{19}] = \_chemical(x_{19})$$
$$\mathbf{T}[h_{22}] = \mathbf{T}[\text{\_poison\_v\_1 ARG0 } e_{24} \text{ ARG1 } x_{19} \text{ ARG2 } x_{23}] = \exists\, e_{24}, \_poison\_v\_1(e_{24}, x_{19}, x_{23})$$
$$\mathbf{T}[h_{25}] = \mathbf{T}[\text{\_the\_q ARG0 } x_{23} \text{ RSTR } h_4 \text{ BODY } h_{28}] = (\exists\, x_{23}, \mathbf{T}[h_4] \wedge \mathbf{T}[h_{28}])$$
$$\mathbf{T}[h_4] = \mathbf{T}[\text{\_the\_q ARG0 } x_6 \text{ RSTR } h_8 \text{ BODY } h_2] = (\exists\, x_6, \mathbf{T}[h_8] \wedge \mathbf{T}[h_2])$$
$$\begin{aligned}\mathbf{T}[h_8] &= \mathbf{T}[\text{compound ARG0 } e_9 \text{ ARG1 } x_6 \text{ ARG2 } x_{10}, \text{\_company\_n\_of ARG0 } x_6 \text{ ARG1 } i_{15}] \\ &= (\mathbf{T}[\text{compound ARG0 } e_9 \text{ ARG1 } x_6 \text{ ARG2 } x_{10}] \wedge \mathbf{T}[\text{\_company\_n\_of ARG0 } x_6 \text{ ARG1 } i_{15}]) \\ &= (\exists\, e_9, compound(e_9, x_6, x_{10}) \wedge \_company\_n\_of(x_6))\end{aligned}$$
$$\mathbf{T}[h_2] = \mathbf{T}[\text{\_ensure\_v\_1 ARG0 } e_3 \text{ ARG1 } x_6 \text{ ARG2 } h_{17}] = (\exists\, e_3, \_ensure\_v\_1(e_3, x_6, \mathbf{T}[h_{17}]))$$
$$\mathbf{T}[h_{17}] = \mathbf{T}[\text{\_no\_q ARG0 } x_{19} \text{ RSTR } h_{21} \text{ BODY } h_{22}] = (\forall\, x_{19}, \mathbf{T}[h_{21}] \rightarrow \neg\mathbf{T}[h_{22}])$$

Figure 5: The transformations of the MRS from Figure 4.

ity to incremental addition of background knowledge. The axioms cover simple lexical semantics gaps such as $\forall x, man\ x \rightarrow person\ x$ and $\forall x, empty\ x \rightarrow \neg full\ x$ that can be easily derived from resources like Wordnet. We also have some axioms related to the ERG abstract predicates, such as $\forall\ e\ x\ y, compound\ e\ x\ y \rightarrow for\ e\ x\ y$[15] and an axiom to deal with the null contribution to the semantics of expletive constructions, $\forall\ x\ y, \_be\_v\_there\ x\ y$.

label	true	false	%
CONTRADICTION	117	213	35
ENTAILMENT	132	198	40
NEUTRAL	330	-	100

Table 2: The results in the SB-SICK. The 'true' means that using MRS Logic, we proved the expected logical relation, and 'false' otherwise. Since neutral is a fallback in our method, we had no error for the neutral label.

We analyzed the cases where we could not prove the expected result, looking for possible translation failures. We summarized some relevant cases we found next, but none were related to problems with translating the MRSs to HOL.

As reported in (Kalouli et al., 2017a,b), we also found that some pairs have wrong labels. For instance, Examples (2) and (3) were annotated as entailment and contradiction, respectively. Example (2) is far from a logical entailment, although somehow related pragmatically. The SICK authors acknowledge these cases as inconsistencies in their dataset.[16]

(2) a. "People are walking outside the building that has several murals on it."
b. "Several people are in front of a colorful building."

(3) a. "The black and white dog is running indoors."
b. "The black and white dog is running in a green yard."

Errors in logical reasoning were also expected since we submitted the HOL formulas to FOL provers, relying on their ability to reduce them to FOL when possible.[17] We also have not implemented axioms to handle all ERG abstract predicates, e.g., nominalization. Some additional background knowledge is undoubtedly necessary and can eventually be induced (Ihsani, 2012), consider formalizing that a guitar player is a guitarist in Example (4).

(4) a. "A person has blonde and flyaway hair and is playing a guitar."
b. "A guitarist has blonde and flyaway hair."

We stressed our transformation rules without finding errors. Second, aligned (Bos, 2014), we validated that the lack of a systematic way to produce relevant background knowledge is the bottleneck of logical inference in RTE.

[15] Remember that compound means an underspecified preposition in the noun-noun compounds. This would be one of many axioms for each possible preposition in English.

[16] Notice that nothing blocks an interpretation of a situation with two distinct dogs or groups.

[17] Some high-order predicates, in the absence of explicit types, can be taken as functions.

5 Conclusion

We presented an open-source library to translate English sentences into HOL formulas. The code is available at http://github.com/ibm/mrs-logic. We tested the library in a dataset of pairs of sentences classified as entailment, contradiction, and neutral. Despite the results, we have collected the necessary insights to refine the RTE procedure and learned that a lot depends on the precise RTE task definition. Fine-grained deep linguistic analyses reveal inconsistencies invisible for purely statistical methods, hiding the real challenge of language understanding.

Considering the most popular approaches for RTE, we differ in using multiple interpretations for each sentence (although limited to four combinations) provided by the grammar-based analyses. We suspect that background knowledge is crucial for selecting the most plausible reading of the sentences when a pair is being tested. Many cases were not proved just because the expected readings of each sentence were not among the tested combinations limited by the computational resources we had.

The literature on computational semantics is vast. We are aware of the range of possibilities from non-compositional representations such as AMR (Banarescu et al., 2013) to inferences directly over surface forms such as Natural Logic (MacCartney and Manning, 2007). We focused on MRS, related to (Lien, 2014), although using logic inference instead of graph matching.

References

Eneko Agirre and Aitor Soroa. 2009. Personalizing PageRank for word sense disambiguation. In *The 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 33–41, Athens, Greece. ACL.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *The 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Emily M. Bender, Dan Flickinger, Stephan Oepen, Woodley Packard, and Ann Copestake. 2015. Layers of interpretation: On grammar and compositionality. In *The 11th International Conference on Computational Semantics*, pages 239–249, London, UK. ACL.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *The 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. ACM.

Luisa Bentivogli, Raffaella Bernardi, Marco Marelli, Stefano Menini, Marco Baroni, and Roberto Zamparelli. 2016. Sick through the semeval glasses. lesson learned from the evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. *Language Resources and Evaluation*, 50:95–124.

Johan Bos. 2014. Is there a place for logic in recognizing textual entailment. *Linguistic Issues in Language Technology*, 9.

Stergios Chatzikyriakidis and Zhaohui Luo. 2020. *Formal Semantics in Modern Type Theories*. Wiley.

Cleo Condoravdi, Dick Crouch, Valeria de Paiva, Reinhard Stolle, and Daniel G. Bobrow. 2003. Entailment, intensionality and text understanding. In *The HLT-NAACL 2003 Workshop on Text Meaning*, pages 38–45.

Ann Copestake. 2002. *Implementing typed feature structure grammars*, volume 110. CSLI publications Stanford.

Ann Copestake and Dan Flickinger. 2000. An open source grammar development environment and broad-coverage english grammar using hpsg. In *The Second Linguistic Resources and Evaluation Conference*, pages 591–600, Athens, Greece.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal recursion semantics: An introduction. *Research on language and computation*, 3:281–332.

Berthold Crysmann and Woodley Packard. 2012. Towards efficient HPSG generation for German, a non-configurational language. In *COLING*, page 695–710.

Leonardo de Moura and Nikolaj Björner. 2008. Z3: An efficient SMT solver. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340, Berlin. Springer.

Guy Emerson. 2020. Linguists who use probabilistic models love them: Quantification in functional distributional semantics. In *The Probability and Meaning Conference (PaM 2020)*, pages 41–52, Gothenburg. ACL.

William M. Farmer. 2008. The seven virtues of simple type theory. *Journal of Applied Logic*, 6(3):267–286.

Dan Flickinger. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28.

Dan Flickinger, Ann Copestake, and Ivan A. Sag. 2000. Hpsg analysis of english. In *Verbmobil: Foundations of speech-to-speech translation*, pages 321–330. Springer, Berlin, Germany.

Michael Wayne Goodman. 2019. A python library for deep linguistic resources. In *2019 Pacific Neighborhood Consortium Annual Conference and Joint Meetings (PNC)*, Singapore.

John Harrison. 2009. HOL Light: An overview. In *Theorem Proving in Higher Order Logics*, pages 60–66, Berlin. Springer.

Allan Hazlett. 2010. The myth of factive verbs. *Philosophy and Phenomenological Research*, 80(3):497–522.

Annisa Ihsani. 2012. Automatic induction of background knowledge axioms for recognising textual entailment. Master's thesis, University of Groningen.

Daniel Jurafsky and James H. Martin. 2023. Speech and language processing. Draft of January 7, 2023.

Aikaterini-Lida Kalouli, Livy Real, and Valeria de Paiva. 2017a. Correcting contradictions. In *The Computing Natural Language Inference (CONLI)*, Montpellier, France.

Aikaterini-Lida Kalouli, Livy Real, and Valeria De Paiva. 2017b. Textual inference: getting logic from humans. In *The 12th International Conference on Computational Semantics (IWCS)*, Montpellier, France.

Alexander Koller and Stefan Thater. 2005. Efficient solving and exploration of scope ambiguities. In *The ACL Interactive Poster and Demonstration Sessions*, pages 9–12, Ann Arbor, Michigan. ACL.

Alexander Koller and Stefan Thater. 2006. An improved redundancy elimination algorithm for underspecified representations. In *The 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 409–416, Sydney, Australia. ACL.

Alexander Koller and Stefan Thater. 2010. Computing weakest readings. In *The 48th Annual Meeting of the ACL*, pages 30–39, Uppsala, Sweden. ACL.

Elisabeth Lien. 2014. Using Minimal Recursion Semantics for entailment recognition. In *The Student Research Workshop at the 14th Conference of the European Chapter of the ACL*, pages 76–84, Gothenburg, Sweden. ACL.

Guilherme Lima, Alexandre Rademaker, and Rosario Uceda-Sosa. 2023. ULKB Logic: A HOL-based framework for reasoning over knowledge graphs. Under review.

Bill MacCartney and Christopher D Manning. 2007. Natural logic for textual inference. In *The ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200.

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 *The Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. ELRA.

George A Miller. 1995. WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Melanie Mitchell. 2023. How do we know how smart ai systems are? *Science*, 381(6654):adj5957.

Leonardo de Moura and Sebastian Ullrich. 2021. The lean 4 theorem prover and programming language. In *The 28th International Conference on Automated Deduction, Virtual Event, July 12–15, 2021, Proceedings 28*, pages 625–635. Springer.

Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. 2002. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*. Springer, Berlin.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. The University of Chicago Press and CSLI Publications, Chicago, IL and Stanford, CA.

Willard Van Quine. 1960. Carnap and logical truth. *Synthese*, 12:350–374.

Aarne Ranta. 1994. *Type-theoretical grammar*. Oxford University Press.

Stephan Schulz, Simon Cruanes, and Petar Vukmirović. 2019. Faster, higher, stronger: E 2.3. In *Automated Deduction – CADE 27*, pages 495–507. Springer.

The Coq Development Team. 2021. *The Coq Reference Manual: Release 8.14.0*.

Dag Westerståhl. 2019. Generalized Quantifiers. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, Winter 2019 edition. Metaphysics Research Lab, Stanford University.

Yoad Winter. 2016. *Elements of Formal Semantics: An Introduction to the Mathematical Theory of Meaning in Natural Language*. Edinburgh University Press.