EasyChair Preprint

№ 217

Graph Knowledge Representations for SICK

Aikaterini-Lida Kalouli, Richard Crouch, Valeria de Paiva and
Livy Real

EasyChair preprints are intended for rapid
dissemination of research results and are
integrated with the rest of EasyChair.

June 1, 2018

Graphical Knowledge Representations for SICK

Aikaterini-Lida Kalouli
University of Konstanz
first.last@uni-konstanz.de

Valeria de Paiva
Nuance

Livy Real
University of São Paulo

Richard Crouch
A9

valeria.depaiva,livyreal,dick.crouch@gmail.com

Abstract

This paper compares three semantic parsing representations in their analysis of a few simple but central phenomena. These are PropS, AMR, and GKR for the representations, and passivization, coordination, and negation for the phenomena.

1 Introduction

Infering entailment and contradiction relations between utterances (texts / sentences / questions and answers) is a necessary, if not sufficient, condition for natural language understanding. Inference is also the central topic of logic, in whatever flavour: philosophical, computational, or mathematical. Semantic parsing of utterances into explicit meaning representations has been one approach employed in recent work on natural language inference, e.g. Abstract Meaning Representation (AMR, (Banarescu et al., 2013)), PropS (Stanovsky et al., 2016), decompositional semantics (Zhang et al., 2017), UDepLambda (Reddy et al., 2017), and Boxer (Bos, 2015). Given the relatively limited amounts of high quality training data for inference, we believe that these explicitly representational approaches have merit viewed alongside end-to-end neural architectures.

A number of larger inference datasets have recently been constructed e.g. SICK citemarelli2014, SNLI (Bowman et al., 2015), MultiNLI (Williams et al., 2017). These datasets are still fairly small by deep learning standards, and it is also not clear how trustworthy they are, nor whether they really measure what humans mean by inference. In (Kalouli et al., 2017b,a, 2018) we investigated the SICK corpus, the earliest and smallest of these corpora. For that we used an easy off-the-shelf open source pipeline that allowed us to uncover several annotation problems and to propose semi-automatic means of discovering and correcting logical mistakes in SICK. In this paper we recall these corrections of mechanical turkers inference annotations, and compare semantic parser meaning representations of selected SICK sentence pairs for three different appoaches: PropS, AMR, and GKR.

Graphical Knowledge Representation (GKR) (Kalouli and Crouch, 2018; Crouch and Kalouli, 2018) is a new open source semantic framework and parser, and can be found at https://github.com/kkalouli/GKR_semantic_parser. It is a descendant of the Abstract Knowledge Representation (AKR) developed at Xerox PARC (Bobrow et al., 2007), but is considerably different in its theory and principal implementation practice. AKR was first decoupled from the XLE/LFG parser by (Crouch, 2014), and then the representation and inference process recast in graphical form in (Boston et al., Forthcoming). GKR, like AKR, has a focus on handling intensional inferences in natural language (hypotheticals, modals, etc) in addition to the more standard fare of first-order inferences. GKR uses enhanced dependencies ((Schuster and Manning, 2016)) obtained via the Stanford Neural Universal Dependency parser (Chen and Manning, 2014) to first create dependency graphs. The dependencies are used as the scaffold from which a number of semantic sub-graphs are constructed, the most important of which are a conceptual, predicate-argument graph, and a contextual graph to deal with Boolean, intensional, and quantificational aspects of meaning. Additional graphs to capture the semantic impact of morpho-syntactic features (e.g. tense, number), and coerference are also built. The design is deliberately modular, so that further application-dependent graphs can be layered in, e.g for distributional word and phrase vectors, or for dialogue state and plan monitoring

(Shen and Harkema, Forthcoming).

In comparing SICK representations across frameworks we do not aim for comprehensiveness, but merely to highlist some of the significant similarities and differences. We also aim to show how GKR fulfills some of the desiderata for a semantic representation set out in (de Paiva, 2011).

2 SICK

SICK (Sentences Involving Compositional Knowledge) by (Marelli et al., 2014) is an English corpus, created to provide a benchmark for compositional extensions of Distributional Semantic Models (DSMs). The data set consists of English sentence pairs, generated from existing sets of captions of pictures. The authors of SICK selected a subset of the caption sources and applied a 3-step generation process to obtain their pairs. This data was then sent to Amazon Turkers who annotated them for semantic similarity and for inference relations, i.e. for entailment, contradiction and neutral stances. Since SICK was created from captions of pictures, it contains literal, non-abstract, common-sense concepts. The corpus is *simplified* in aspects of language processing not fundamentally related to compositionality: there are no named entities, the tenses have been simplified to the progressive only, there are few modifiers, few pronouns etc. The curators of the corpus also made an effort to reduce the amount of encyclopedic world-knowledge needed to interpret the sentences.

The number of sentences pairs of the corpus may seem substantial (almost 10K of pairs), but there is much redundancy in the contents. Due to the construction process many of the sentences are repeated in different pairs. In total there are 6076 unique sentences and only some 2000 unique lemmas whose meanings are to be found in Princeton WordNet synsets. The data set consists of 9840 sentence pairs, which have been annotated as

1424	pairs of contradictions	$(AcBBcA)$
1300	pairs of double entailment	$(AeBBeA)$
1513	pairs of single entailment	$(AeBBnA)$
4992	pairs of neutrals	$(AnBBnA)$

The SICK corpus is a good dataset to test approaches to semantic representations and natural language inference, due to its intended, human-curated simplicity. The pairs mostly talk about everyday, concrete actions and actors. The sentences are short; there are no complicated syntactic structures. There are very few ellipsis and the sentences are mostly grammatical and short. There was also an effort by the corpus creators to remove all named entities from sentences in SICK, which is helpful for the tasks we want to focus on. Additionally, there were efforts to limit the number of compounds and pronouns and tenses were converted to the progressive only.

2.1 Previous Work on SICK

About 18 months ago we started to investigate SICK to make sure it was a **trustworthy** baseline for work in inference. However, we soon discovered that the data is very noisy, so we needed to provide mechanisms to produce **correct** annotations.

In a series of papers we discussed ways of correcting the corpus SICK ((Kalouli et al., 2017b),(Kalouli et al., 2017a), (Kalouli et al., 2018)). As we argued, the inferences obtained from the several existent inference-based corpora (Bowman et al., 2015; Williams et al., 2017) need to match humans' intuitions of logical inference. Hence, we should not have asymmetric contradictions (A is contradictory to B, but B is neutral with respect to A) nor should we have inferences that contradict common sense (such as a *flute* entails a *guitar*). Nevertheless this is what happens with several of the original SICK annotations. This is why we needed to correct such mistakes, so that we could use SICK as our baseline corpus for natural language inference.

To do our corrections and analyses, we first put together an easy off-the-shelf pipeline. The reasoning was that we would like to know how many of the SICK inferences could be done simply using open source resources like WordNet and SUMO. We used CoreNLP (Schuster and Manning, 2016) to produce enhanced Universal Dependencies, We used JIGSAW(Basile et al., 2007) for disambiguation into Princeton WordNet (PWN) senses, then we used PWN to SUMO mappings to associate a bag of SUMO concepts to each sentence[1]. Using this basic data, we first checked the collection of inferences annotated by the turkers as *single entailments*, the 1513 pairs $AeBBnA$. We discovered that 12% of these pairs needed corrections and we supplied these (Kalouli et al., 2017a).

[1] The processed corpus data is available from GitHub https://github.com/kkalouli/SICK-processing

Secondly we investigated the contradictions in the corpus and discovered that as many as 611 pairs were actually 'asymmetric contradictions'. Since the turkers were asked to annotate only one direction entailments, in as many as 611 pairs, the annotators thought that A was contradictory to B, but B was neutral to A. And in a few cases they even had that B entailed it! This meant that around 30% of all contradictions annotated in the corpus were non-logical. To correct these we needed to tighten the guidelines much (Kalouli et al., 2017b). As discussed by the SICK creators themselves, the lack of common referents was a big issue when dealing with contradictions. (This was also discussed when producing the SNLI corpus (Bowman et al., 2015).) But there were other issues too, like the use of *privative* adjectives and nouns (for instance a *fake gun* is not a *gun*, the same way a *cartoon airplane* is not an *airplane*), etc. We decided that we can only detect contradictions in sentences that are 'close enough'. In particular there are predicates 'contradictory in context', that need commonsense to be detected. For example *Children in red shirts are playing in the leaves* and *Children in red shirts are sleeping in the leaves* need to be annotated as a contradiction, although *sleep* and *play* are not direct antonyms.

Thirdly we came up with one approach to automatically annotating and correcting the inference pairs in SICK, based on the observation that several SICK pairs differ only by one word. Differing by "one word" means that there is either one more word in the one sentence than in the other or that each of the sentences contains a word that is not found in the other one. These pairs that differ only by one word (or none) are what we called ”easy inferences” in (Kalouli et al., 2018). Using this approach we could automatically correct and re-annotate some of the pairs without having to solve all the inference challenges associated with the meanings of the sentences first. We end up with 2936 (almost one third of the corpus) pairs being "one-word apart" and for these we can mostly infer from the relationship in WordNet between the words apart, the relationship between the sentences, see the details in the paper. This is clearly one of the drawbacks of the elicitation process used to produce the inference corpus. When humans are asked to contradict or infer from one sentence another, they tend to make the minimal modification necessary for the goal at hand, many times changing one single word.

3 Graphical Knowledge Representation (GKR)

As mentioned in the introduction, despite some logical similarities between GKR and AKR, in particular their use of concepts and contexts, their sources are very different: GKR uses enhanced dependencies (Schuster and Manning, 2016) obtained via the Stanford Neural Universal Dependency parser (Chen and Manning, 2014) to create dependency graphs, on top of which fuller semantic graphs are constructed by GKR. On the other hand, AKR, the semantic component of the Xerox Language Engine (XLE) platform is based on Lexical Functional Grammar (LFG) to create the representations. The AKR approach was decoupled from XLE/LFG in (Crouch, 2014) and then revisited in an explicitly graphical form in (Boston et al., Forthcoming), recasting AKR as a set of layered sub-graphs, including a conceptual graph, a contextual graph, a property graph, a syntactic dependency graph, a co-reference graph, and with the possibility of layering in further sub-graphs should an application demand it.

GKR is a semantic parser that rewrites a given sentence into a layered semantic graph. The implementation of the parser is done in Java. The semantic graph is a rooted, node-labelled, edge-labelled, directed graph. It consists of at least four sub-graphs, layered on top of a central conceptual (or predicate-argument) sub-graph. Each such graph encodes different information and thus the parser is highly modular. There is a dependencies sub-graph, a conceptual sub-graph, a contextual sub-graph, a properties sub-graph and a lexical sub-graph.

The Stanford CoreNLP software is used to produce the enhanced++ dependencies of Schuster and Manning (2016). The output dependencies are straightforwardly rewritten to the GKR dependency graph. The conceptual graph, produced next, contains the basic predicate-argument structure of the sentence: what is talked about; the semantic subject or agent, the semantic object or patient, the modifiers, etc.

The conceptual graph is the core of the semantic graph and glues all other sub-graphs together. Note that, as discussed in (Kalouli and Crouch, 2018) and (Condoravdi et al., 2001), the variables in the conceptual graph are interpreted as con-

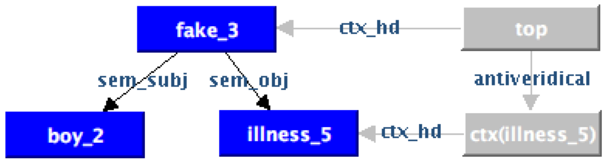

Figure 1: The conceptual (left) and the contextual (right) graph of *The boy faked the illness.*

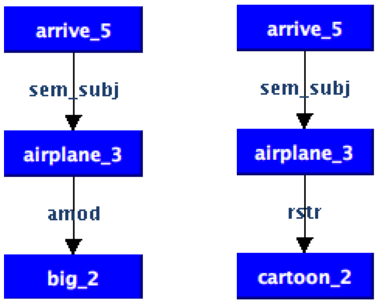

Figure 2: The conceptual graphs of *The big airplane is arriving* (left) and *The cartoon airplane is arriving* (right).

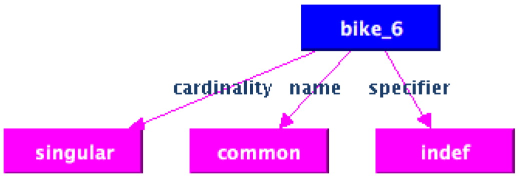

Figure 3: Part of the properties graph of *Two people are riding a bike.*

cepts, not collections of individuals. Also the conceptual graph does not make any commitments about the existence or otherwise of those concepts. In Figure 1 in blue we see the conceptual graph of the sentence *The boy faked the illness*: the predicate is the root node and the semantic subject and object are children nodes. No further existential information is encoded in this graph so no judgments about truth or entailment can be made so far. However, the graph does allow judgments about semantic similarity. In fact, it also accounts for hard cases like the ones with privative adjectives mentioned before. The conceptual graphs of the sentences *A* = *The big airplane is arriving*, *B* = *The cartoon airplane is arriving* are shown in Figure 2. The graph of sentence A modifies the concept of *airplane* so that a subset of the concept is addressed, while sentence B imposes a restriction on *airplane* so that a new concept *cartoon airplane* is involved. If the two graphs were to be compared for their semantic similarity, they would correctly be judged non-similar despite the phenomenical closeness of their basic concepts.

The contextual graph provides the existential commitments of the sentence. It introduces a top context (or possible world) which represents whatever the author of the sentence takes the described world to be like; in other words, whatever her "true" world holds. Additional contexts can be introduced, corresponding to any alternative, possible worlds introduced in the sentence. Such contexts can be introduced by negation, disjunction, modals, clausal contexts of propositional attitudes (e.g. belief, knowledge, obligation), implicatives and factives, imperatives, questions, conditionals and distributivity. The contextual graph is built on top of the concepts graph, as shown in Figure 1 on the right. For our example the graph gives us the information that there is a top context in which only the concept (*fake*) linked with the *ctx_hd* arc is instantiated. The top context contains the embedded context of *illness*, in which again only the concept of *illness* is instantiated through the *ctx_hd* arc. But the context of *illness* is antiveridical in the top context: it does not exist in it which is accurate as the illness was faked.

The properties graph associates the conceptual graph with morphological and syntactical features such as the cardinality of nouns, the verbal tense and aspect, the finiteness of specifiers, etc. The extracted properties can then be imposed on the corresponding concepts as restrictions to compute inference or solve issues like distributivity mentioned before. This can be seen in Figure 3, which depicts only a part of the properties graph of the sentence *Two people are riding a bike*; it shows the properties of the concept *bike*. The cardinality of the noun is set to *singular* but its specifier to *indefinite* so that when the two restrictions are combined – e.g. for doing inference –, the underspecificity required is kept intact: there is not necessarily one bike involved, but also not two; both readings are still allowed.

Finally, the lexical graph carries the lexical information of the sentence. It associates each node of the conceptual graph with its disambiguated sense and concept, its hypernyms and its hyponyms, making use of the disambiguation algorithm JIGSAW (Basile et al., 2007), WordNet (Fellbaum, 1998)) and the knowledge base SUMO (Niles and Pease, 2001). To build the lexical graph, the whole sentence is first run through the JIGSAW algorithm which disambiguates each word of the sentence by assigning it the WordNet sense with the highest probability. Then, the disambiguated sense is matched to its WordNet hyponyms and hypernyms and to the SUMO concepts. This means that the tokens of the sentence are expanded to other senses and concepts, in a way that allows their look-up and their interlinking. This idea is integrating our insight that lexical resources are not enough for inference systems and that "smarter", state-of-the-art methods have to be infused: word vectors can easily be added to the graph so that the existing words are further expanded and inter-linked. With that we are aiming at a hybrid parser, which makes use of the strengths of both methods: it relies heavily on rules for the graph construction but also integrates machine-learning output for the graph expansion.

The implemented parser rewrites a given sentence to the subgraphs discussed and with that, it produces the final semantic graph for the sentence. This representation is suitable for further processing into question-answering or dialog systems, into inference – as it will be shown shortly – and other semantic tasks.

4 Semantic Representations

Given the much advertised new capabilities of neural net dependency parsers and POS-taggers and the linguistically informed simplifications of the SICK corpus, it seemed reasonable to see how many of these sentences get a reasonable shallow semantic representation and how these shallow interpretations can help with inference. But there is now a proliferation of semantic representations, as hinted in the introduction. Which semantic representation should we choose and why?

To help us decide we choose a very small collection of SICK pairs and tried to apply three semantic representations to them: ProPs, AMR and GKR. This is not an empirical study, simply a way of getting more information about the space of possibilities. This also helps to clarify the important differences between these semantic possibilities. Since the pairs of SICK are syntactically simplified, for this work we decided to choose pairs dealing with the most complex phenomena involved in SICK, with the assumption that if these phenomena work good enough in the various representations, the representations will also be able to deal with the simpler phenomena. Therefore, we chose pairs involving passive voice, coordination, and negation. These are:

- A: *A guitar is being played by the man.*
  B: *A man is playing a trumpet*

- A: *A dog is running on concrete and is holding a blue ball.*
  B: *A dog is running on concrete and is holding a ball.*

- A: *There is no cyclist performing a jump on a bicycle.*
  B: *A cyclist is performing a jump on a bicycle.*

There are a number of other phenomena not showcased in SICK, such as clausal complements, imperatives, conditionals and implicatives/factives, for which it is really interesting to compare the various representations but these comparisons will have to wait for a future work. Additionally, the pairs we chose involve inference relations which raise interesting discussion points both about the most suitable semantic representation to compute inference as well as about the required inference. In the following we will discuss how our chosen pairs behave in each of the three representations.

4.1 PropS

The PropS system (Stanovsky et al., 2016) is designed to explicitly express the propositional structure of a sentence. The system abstracts away from the syntactic structure by adding relations such as *and* for coodination or *outcome* and *condition* for conditionals or *prop_of* for relative clauses, all necessary semantic notions to be extracted from a given sentence. This means that it represents such phenomena more explicitly than the dependencies do, as we can also observe by running a couple of SICK pairs with it. However, by addressing only propositional / clausal structure, PropS does not provide sufficient information about the internal semantics of many noun phrase to drive a number of SICK inferences

Passives

PropS representations normalize passive and active alternations:

```
played:(subj:the man , obj:a guitar )
playing:(subj:A man , dobj:a trumpet )
```

However, as can be seen there is no normalization of the verb inflections.

Coordination

The sentences of the pair *A= A dog is running on concrete and is holding a blue ball. B= A dog is running on concrete and is holding a ball.* are separated in their component clauses and additionally there is a third relation added to connect the two clauses with the *and* relation and to also make the subject of the coordinated clause explicit. Sentence A becomes

```
running:(subj:A dog , prep_on:concrete)
holding:(subj:A dog , dobj:a blue ball )
and:(conj_and:A dog is running ... and ,
     conj_and:A dog is holding ...)
```

Is this complementary relation really useful in the way it is presented? If we run the two sentences but with disjunction instead of conjunction, i.e. *A dog is running on concrete or is holding a blue ball.*, we will observe a similar kind of representation: the two clauses are successfully separated and there is a third complementary relation added which connects the two clauses with *or*:

```
holding:(subj:A dog , dobj:a ball )
running:(subj:A dog , prep_on:concrete )
or:(conj_or:A dog is running ... or ,
    conj_or:A dog is holding a ball)
```

This means that the representations for disjunction and conjunction look alike except for the *or* and *and*, which remain “un-decoded” into some deeper semantic notion. Ideally, we would like the representation to show us that in the first pair the two clauses co-occur and are both true while in the second pair only one of the clauses can occur and be true. For this specific pair the inference relation is not influenced by the presence of coordination as the inference boils down to the easy entailment *blue ball* → *ball*. If, however, sentence B of the pair were the sentence *The dog is holding a ball*, it would make a difference if there is conjunction or disjunction in A and if the representation can reliably show us what contexts are instantiated in which case.

Negation

We can look at the third and probably most complex phenomenon of the ones we chose: negation. It seems that for such more complex phenomena of natural language ProPs is not offering a more complete semantic structure. This is the case with the pair A = *There is no cyclist performing a jump on a bicycle.* B = *A cyclist is performing a jump on a bicycle*. For the first sentence we get the representation

```
Exists:(subj:no cyclist performing
               a jump on a bicycle )
```

and for the second one the following:

```
performing:(subj:A cyclist,
            dobj:a jump on a bicycle )
```

Sentence A encodes existence but in a insufficient way because *no cyclist* is claimed to *exist*. But if there is no cyclist, he/she does not exist. The representation is not dealing with negation in a usable way for semantic tasks because it is left unprocessed in a way that it almost conveys a counter-to-logic notion. If we tried to compute the inference relation between the two pairs, we would have to do our own post-processing of the representations to account for the fact that in A the cyclist does not exist while in B it does. The sentences could not be compared directly.

4.2 AMR

Abstract Meaning Representation (AMR) is a relatively recent semantic representation (Banarescu et al., 2013) where the meaning of a sentence is encoded as a rooted, directed graph. The representation is based on manual annotation of the structures and is thus expensive. The automatic creation of AMRs (AMR parsing) has been strongly pursued, however AMR parsing accuracy is still in the high 60%, as measured by the SMatch score, and a significant improvement is needed in order for it to positively impact a larger number of applications (Flanigan et al., 2014; Wang and Xue, 2017). Standard AMR ignores function words, tense, articles, plurality, and prepositions which means that some important information for the semantic processing remains unavailable. Additionally, AMR has limited expressive power for universal quantification and negation (Bos, 2016; Stabler, 2017) and does not make a distinction between real and *irrealis* events (in the example *The boy faked the illness.* the representation commits to the existence of a non-existing illness). For some engineering applications, tense, plurality and quantification may not matter, but for other applications it is obviously important.

Abstract Meaning Representation (AMR) is strongly biased towards English, as pointed out in the original report, and annotation efforts have mostly focused on English. However, in order to train parsers on other languages, methods based on annotation projection, which involves exploiting annotations in a source language and a parallel corpus of the source language and a target language have been used (Damonte and Cohen, 2017; van Noord et al., 2018). We used the Damonte et al online demo http://cohort.inf.ed.ac.uk/amreager.html to check our understanding of the system, but removed their alignments, and also corrected details of the representations if they were obviously incorrect.

Passive

For the easy passive voice examples AMR delivers the desired result, the graphs for the pair

```
# ::snt A guitar is being played
        by the man
(v2 / play-08
    :ARG1 (v1 / guitar)
    :ARG0 (v3 / man))

# ::snt A man is playing a trumpet
(v2 / play-08
    :ARG0 (v1 / man)
    :ARG1 (v3 / trumpet))
```

As can be seen, the AMR normalizes the passive alternation, so that the man is the ARG0 and the instrument the ARG1 in both cases. Whether the pairs contradict depend on assumptions about (a) whether reference is made to the same man at the same time, which is enforced by SICK, and (b) whether it is possible to play two instruments at once.

Coordination

The first sentence in the coordination pair gets:

```
# ::snt A dog is running on concrete
        and is holding a blue ball .
(v4 / and
    :op1 (v2 / run-01
        :ARG0 (v1 / dog)
        :location (v3 / concrete))
    :op2 (v5 / hold-01
        :ARG1 (v7 / ball
            :mod (v6 / blue))
        :ARG0 v1))
```

The second is the same, other than missing the blue modifier on ball. Graph subsumption, whereby a more specific graph entails a more general one would give the expected entailment relation. However, as we will see, and as should hopefully be obvious anyway, subsumption on AMR graphs is not an appropriate way of detemining inference relationships.

Exchanging *and* for *or* produces an AMR that differs only in the coordination operator at the root of the graph. Without further interpretive rules, there is no way of differentiating the entailments of conjunction and disjunction; namely that for a conjunction both conjuncts must be true, whereas for a disjunction at least one must be. Such an interpretaive rule would be to transform the AMR into a more conventional logical formula, perhaps along the lines of (Bos, 2016) or (Stabler, 2017), where the coordination operators are translated to their corresponding propositional connectives.

Negation

For negation we get

```
# ::snt There is no cyclist performing
        a jump on a bicycle .
(v1 / cyclist
      :polarity -
      :ARG0-of (v2 / jump-07
            :location (v3 / bicycle)))

# ::snt A cyclist is performing
        a jump on a bicycle .
(v2 / jump-07
    :ARG0 (v1 / cyclist)
    :location (v3 / bicycle))
```

Both AMRs treat *perform* as a kind of light verb, simplifying *perform a jump* to *jumping*. The first AMR also makes use of an inverted role: the cyclist is the ARG0 of the jump, but the order of arguments is inverted. It is not entirely clear whether inverted roles are pure syntactic sugar, to ease the annotation task for relative clause and other verbal noun modifications like *cyclist performing a jump*. If so, de-inverting the role would give:

```
There is no cyclist performing
a jump on a bicycle .
(v2 / jump-07
    :ARG0 (v1 / cyclist
           :polarity -)
    :location (v3 / bicycle))
```

This would seem to imply that there is a jumping on a bicycle, but just not one being performed by a cyclist. This certainly describes one situation under which the sentence would be true, but not the only one: there could also be no jumping at all, or no bicycles involved. It would be a push to describe these three distinct scenarios as being different interpretations or readings of the sentence. Rather there is just one interpretation (it is not the case that a cyclist is peforming a jump on a bicycle), and three things that could be missing to make it true (no jumping, not by a cyclist, not on a bicycle).

The normalized AMR contradicts the second sentence only under SICK-specific interpretive constraints that the sentences/captions describe the only significant state of affairs depicted in a picture. One says that there is just one jump on a bicycle, but not being performed by a cyclist, whereas the other says that this jump is being performed by a cyclist. However, the contradiction should be stronger than that: under no reasonable interpretive conditions can both sentences be true.

It would therefore appear that role inversion needs to be more complicated than argument inversion. The desired, strong, contradiction can be mantained if, while de-inverting the arguments, and polarity marking remain at their initial level within the AMR, thus normalizing the first sentence to

```
There is no cyclist performing
a jump on a bicycle .
(v2 / jump-07
    :polarity -
    :ARG0 (v1 / cyclist)
    :location (v3 / bicycle))
```

But even here, it is still apparent that AMR inference cannot be reduced to simple graph subsumption. The positive AMR is the same as the negative other than the presence of polarity marker. Under the assumption that more specific graphs entail more general ones, this would mean that the positive entails the negative. So, inference requires first translation AMRs to logical formulas, where the negative polarity is cashed out as a boolean operator taking scope over a certain formula.

4.3 GKR

To ease comparison, we will (a) use the string formatted rather than the graphical version of GKR output, and moreover (b) reformat it using an AMR-style representation of graphs. This involves writing the concept and context graphs as two separate graphs, using concept variables to link them. The property graph is folded into the concept graph as semantically relevant morphosyntactic features associated with concept nodes.

Passive

For *A guitar is being played by the man.* we get:

```
(play_5 / play
    :tense present
    :sem_subj (man_8 / man
          :cardinality singular
          :specifier definite)
    :sem_obj (guitar_2 / guitar
          :cardinality singular
          :specifier indefinite))

(t / _context
    :head play_5
    :introduces man_8, guitar_2
    :relative_polarity veridical)
```

Aside from the additional tense, cardinality, and specifier features, the concept graph is parallel to that for AMR. The difference lies in the context graph, which in this case is not very interesting. It asserts that all the concepts in the concept graph are instantiated within a veridical, top-level context. As with AMR and PropS, the passive and active alternations are normalized to make them similar.

Coordination

For *A dog is running on concrete and is holding a blue ball*, with most of the property graph ommited for brevity, we get:

```
(and_7 / _group
     :sem_subj (dog_2 / dog)
     :is_element (hold_9 / hold
           :sem_obj(ball_12 / ball
               :amod(blue_11 / blue)))
     :is_element (run_4 / run
               :pmod (concrete_6 / concrete
                     :pspec on)))

(t / _context
    :head and_5
    :introduces dog_2, hold_9, ball_12,
               blue_11, run_4, concrete_6
    :relative_polarity veridical)
```

Once again the concept graph is similar to the full AMR, the differences being that the subject is not distributed over the conjuncts, and the prepositional specifier of the locative modifier *on concrete* is preserved in the property graph. The *_group* concept can be viewed as the union of the running and holding concepts.

As with the passivization examples, the context structure here is not very interesting. However, this changes if *and* is replaced by `or`. The concept structure remains unchanged, but two additional disjunctive contexts are introduced

```
(t / _context
    :head or_5
    :disjunction (or_1 / _context
        :head hold_9
        :introduces ball_12, blue_11
        :relative_polarity averidical)
    :disjunction (or_2 / _context
        :head run_4
        :introduces concrete_6
        :relative_polarity averidical))
```

The disjunction relation between the higher level context *t* and the disjunct context *or_1* imposes an averidical relative polarity: the head concept of *or_1* may or may not be instantiated. However, at least one of the two disjunct heads needs to be instantiated. With that we get a useful distinctive modeling of conjunction and disjunction, which can directly be used in further semantic tasks. As was pointed out before, the accurate representation of such semantic phenomena is of great importance in cases where the inference relation within a pair is based on the co-existence or not of different concepts.

Negation

The role of GKR's context structure becomes more apparent with negation. For *There is no cyclist performing a jump on a bicycle* we get:

```
(be_2 / be
  :tense present
  :sem_subj (cyclist_4 / cyclist
    :cardinality no
    :rel_subj (perform_5 / perform
        :sem_obj(jump_7 / jump
            :pmod (bicycle_10 / bicycle
                   :pspec on)))))

(t / _context
  :head be_2
  :relative_polarity veridical
  :not (c1 / _context
    :head cyclist_4
    :introduces perform_5, jump_7,
                bicycle_10
    :relative_polarity anti_veridical)
```

The expletive-be construction is not eliminated from the concept structure in the way that AMR does; this is to support the tense distinction between *there is* and *there was*. The progressive participle modification of *cyclist* is treated along the lines of a reduced relative clause: c.f. *cyclist performing a jump* and *cyclist that is performing a jump*. This introduces an (inverted) relative_subject role, which indicates that *cyclist_4* is both modified by *perform_5*, but also acts as its subject. Apart from the cardinality property on *cyclist_4*, the concept structure marks no difference between the negative and positive versions of the sentence (*There is a cyclist performing a jump on a bicycle*).

The difference is marked in the context structure, where the determiner *no* introduces a second anti-veridical context *c1* whose head is *cyclist_4*. Within the context *c1* the head concept *cyclist_4* of a cyclist performing a jump on a bicycle has an instance, for which to be true there must also be an instance of a jump and a bicycle. However, *c1* is anti-veridical with respect to the top level, speaker committment, context *t*, which means that the concept is asserted to be uninstantiated in *t*. The non-instantiation of *cyclist_4* does not rule out the presence of any jumps, bicycles, or cyclists; only that if any jumps on bicycles are going on then they are not being done by cyclists. The positive version of the sentence, on the other hand, places *cyclist_4* in a veridical context. This leads to a direct contradiction between the two, since one claims that *cyclist_4* is instantiated while the other claims it is uninstantiated.

5 Conclusions

There has been a resurgence of meaning representation languages, spear-headed perhaps, by AMR and the multiple projects using it. Despite the large amount of work on this simplified representation, its semantics does not seem to have the same level of consensus amongs researchers as does the syntax of Universal Dependencies. The different semantic representations seem more different amongst themselves.

Bos (Bos, 2016) and Stabler(Stabler, 2017) want to see AMR graphs as fragments of First-order Logic or Higher-Order Logic, respectively, via translations and propose to augment AMR along these lines. Enhanced Dependencies++ (Schuster and Manning, 2016) and ProPs(Stanovsky et al., 2016) instead want to extend the reach of the syntactc annotations, without aiming for a fully semantic representation. The representation formalisms described as scoped DRS in (van Noord et al., 2018) seems the closest to the GKR we advocate in this paper. Their boxes are like contexts, but they seem to insist on a semantics based on individuals, instead of subconcepts. While we share the use of contexts to deal with modal notions, it is not clear to which extent they see their use of nested boxes as going beyond first-order logic or not.

The work on GKR is only starting, hence temporal phenomena, coreference resolution and implicative behaviour, for example are, so far, only stubbed. But the data in SICK is simplified exactly along these dimensions. Thus we expect SICK representations to be very much equivalent in the three systems we compare. The representation of negation is very different though: the AMR graphs for *No dog is emerging from a lake* and *There is no dog emerging from a lake* are not the same.

References

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 *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186.

Pierpaolo Basile, Marco de Gemmis, Anna Lisa Gentile, Pasquale Lops, and Giovanni Semeraro. 2007. UNIBA: JIGSAW algorithm for word sense disambiguation. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 398–401, Prague, Czech Republic. Association for Computational Linguistics.

Daniel G Bobrow, Bob Cheslow, Cleo Condoravdi, Lauri Karttunen, Tracy Holloway King, Rowan Nairn, Valeria de Paiva, Charlotte Price, and Annie Zaenen. 2007. Parc's bridge and question answering system. In *Proc. of the GEAF 2007 Workshop. CSLI Studies in Computational Linguistics Online*.

Johan Bos. 2015. Open-domain semantic parsing with boxer. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*.

Johan Bos. 2016. Expressive power of abstract meaning representations. *Computational Linguistics*, 42(3):527–535.

Marisa Boston, Richard Crouch, Erdem Ozcan, and Peter Stubley. Forthcoming. Natural language inference using an Ontology. In *Lauri Karttunen Festschrift*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750.

Cleo Condoravdi, Dick Crouch, John Everett, Valeria Paiva, Reinhard Stolle, Danny Bobrow, and Martin van den Berg. 2001. Preventing existence. In *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*, pages 162–173. ACM.

Richard Crouch. 2014. Transfer semantics for the Clear parser. In *Proceedings of NLCS 2014*, page Transfer Semantics for the Clear Parser.

Richard Crouch and Aikaterini-Lida Kalouli. 2018. Named graphs for semantic representations. In *Proceedings of *SEM 2018*.

Marco Damonte and Shay B. Cohen. 2017. Cross-lingual abstract meaning representation parsing. *CoRR*, abs/1704.04539.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.

J. Flanigan, S. Thomson, J. Carbonell, C. Dyer, and N. A. Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proc. of ACL*, Baltimore, Maryland. Association for Computational Linguistics.

Aikaterini-Lida Kalouli and Richard Crouch. 2018. GKR: the graphical knowledge representation for semantic parsing. In *Proceedings of SemBEAR 2018*.

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

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

Aikaterini-Lida Kalouli, Livy Real, and Valeria de Paiva. 2018. Wordnet for "easy" textual inferences. In *Proceedings of the Globalex Workshop, LREC 2018*.

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 LREC 2014*.

Ian Niles and Adam Pease. 2001. Toward a Standard Upper Ontology. In *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, pages 2–9.

Rik van Noord, Lasha Abzianidze, Hessel Haagsma, and Johan Bos. 2018. Evaluating scoped meaning representations. *CoRR*, abs/1802.08599.

Valeria de Paiva. 2011. Bridges from language to logic: Concepts, contexts and ontologies. *Electronic Notes in Theoretical Computer Science*, 269:83 – 94. Proceedings of the Fifth Logical and Semantic Frameworks, with Applications Workshop (LSFA 2010).

Siva Reddy, Oscar Täckström, Slav Petrov, Mark Steedman, and Mirella Lapata. 2017. Universal semantic parsing. *arXiv preprint arXiv:1702.03196*.

Sebastian Schuster and Christopher D Manning. 2016. Enhanced english universal dependencies: An improved representation for natural language understanding tasks. In *LREC*.

Jiaying Shen and Hendrik Harkema. Forthcoming. Layered semantic graphs for dialogue management.

Edward Stabler. 2017. Reforming amr. In *International Conference on Formal Grammar*, pages 72–87. Springer.

Gabriel Stanovsky, Jessica Ficler, Ido Dagan, and Yoav Goldberg. 2016. Getting more out of syntax with props. *arXiv preprint arXiv:1603.01648*.

Chuan Wang and Nianwen Xue. 2017. Getting the Most out of AMR parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1257–1268.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *CoRR*, abs/1704.05426.

Sheng Zhang, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. 2017. Ordinal common-sense inference. *Transactions of the Association for Computational Linguistics*, 5:379–395.