

GKR: the Graphical Knowledge Representation for semantic parsing

Aikaterini-Lida Kalouli

University of Konstanz

aikaterini-lida.kalouli@uni-konstanz.de

Richard Crouch

A9.com

dick.crouch@gmail.com

Abstract

This paper describes the first version of an open-source semantic parser that creates graphical representations of sentences to be used for further semantic processing, e.g. for natural language inference, reasoning and semantic similarity. The Graphical Knowledge Representation which is output by the parser is inspired by the Abstract Knowledge Representation, which separates out conceptual and contextual levels of representation that deal respectively with the subject matter of a sentence and its existential commitments. Our representation is a layered graph with each sub-graph holding different kinds of information, including one sub-graph for concepts and one for contexts. Our first evaluation of the system shows an F-score of 85% in accurately representing sentences as semantic graphs.

1 Introduction

Semantic parsing to construct graphical meaning representations is an active topic at the moment (Banarescu et al., 2013; Perera et al., 2018; Flanagan et al., 2014; Wang et al., 2015; Berant et al., 2013). It is not without its critics, however. Bender et al. (2015) object to the conflation of sentence meaning with speaker meaning, inherent in trying to use annotations to learn a direct mapping from sentences onto highly domain specific meaning representations. Bos (2016) and Stabler (2017) have also questioned the expressive power of Abstract Meaning Representation (AMR) (Banarescu et al., 2013), one of the most popular graphical meaning representations.

We believe that both lines of criticism are well-founded, but that there is still value in parsing to produce graphical representations. This paper describes the first version of an open source semantic parser that creates graphical representations that are inspired by those produced by the proprietary system described in Boston et al. (forthcoming). Salient features of the system are:

- It uses the enhanced dependencies (Schuster and Manning, 2016) of the Stanford Neural Universal Dependency parser (Chen and Manning, 2014) to create dependency graphs, on top of which fuller semantic graphs are constructed.
- Interaction between different sub-graphs is used to account for phenomena like Booleans (negation, disjunction), modals and *irrealis* contexts, distributivity and quantifier scope, co-reference, and sense selection.
- Though oriented to using formal ontologies to support a Natural Logic (MacCartney and Manning, 2007) style of Natural Language Inference (NLI), it also supports the somewhat different task of measuring semantic similarity.
- More philosophically, we view our graphs as first-class semantic objects that should be directly manipulated in reasoning and other forms of semantic processing. We do not see them as just a prettier way of writing down formulas in first- or higher-order logic.

In the next section we briefly describe the precursors and motivations behind our approach. In section 3 we present the Graphical Knowledge Representation (GKR) and how it is constructed. Section 4 evaluates the current parsing into GKR, while section 5 discusses our future additions to the system. In section 6 we compare GKR to other similar representations and parsers. In the last section we offer our conclusions and point to a companion paper discussing named graphs.

2 AKR and Layered Graphs

The so-called Abstract Knowledge Representation (AKR)¹ (Bobrow et al., 2007b,a) focused on in-

¹AKR is the semantic component of the XLE platform (Maxwell and Kaplan, 1996)

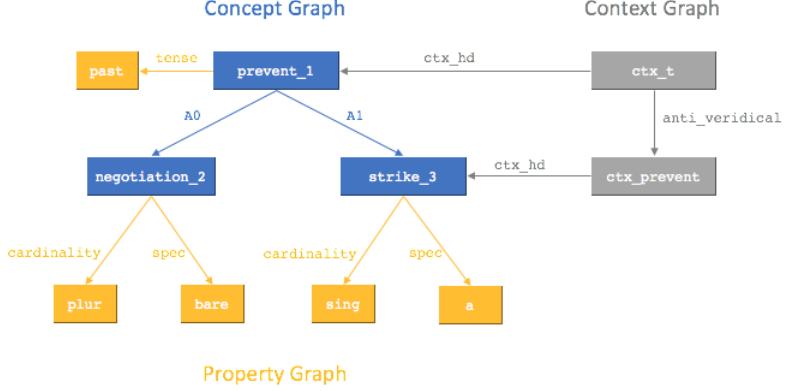


Figure 1: Concept graph (blue), property graph (yellow) and context graph (grey) from Boston et al. (forthcoming) for *Negotiations prevented a strike*.

tensional phenomena in natural language, with the sentence *Negotiations prevented a strike* being a driving example (Condoravdi et al., 2002). The claim was that, viewed in the right way, the logical formula

$$\exists n, s. \text{negotiation}(n) \wedge \text{strike}(s) \wedge \text{prevent}(n, s)$$

was a correct but incomplete semantic representation. It is correct if the variables n and s are construed as referring to sub-concepts of the concepts *negotiation* and *strike*, rather than to an individual strike or negotiation. The formula just describes the subject matter: some kind of prevention, restricted to a relation between some kind of negotiation and some kind of strike. The formula, construed as talking about concepts, makes no assertions about the existence or otherwise of any such negotiations or strikes. To complete the representation it is necessary to add a contextual level that makes assertions about whether instances of the concepts exist. In this case there are two contexts. A top level context in which the negotiation concept is asserted to have an instance; and a hypothetical (prevented) context in which the strike is claimed to have an instance. The two contexts are in an anti-veridical relationship, meaning that the *strike* concept that has an instance in the lower hypothetical context has no instance in the top context. Later work (Nairn et al., 2006) used this framework to capture a wide variety of relative polarity inferences arising from factive and implicative verbs.

A semantics for a variant of AKR was presented in the form of a Textual Inference Logic (TIL)

(de Paiva et al., 2007). This recast AKR as a contexted description logic, but was not strictly faithful to AKR’s eschewal of reference to individuals in favor of reference to concepts. The underlying semantics for TIL followed that of description logic by not taking concepts as primitive, but instead defining concept relations in terms of relations between sets of individuals in concept extensions.

The approach was revisited in an explicitly graphical form (Boston et al., forthcoming), recasting AKR as a set of layered sub-graphs, including a conceptual graph, a contextual graph, along with a property graph, syntactic dependency graph, a co-reference graph, and with the possibility of layering in further sub-graphs should an application demand it. The graphical representation of *Negotiations prevented a strike* is shown in Figure 1.

The graphical format was more than just notational sugar to provide more colorful and accessible representations. First, dominance in the concept and property graphs is strictly aligned with concept restriction: the parent concept is subsequently restricted by the child concept or property. Second, a strict separation between the concept and context graph is enforced: concepts cannot be restricted by contexts. Just one kind of link between contexts and concepts is permitted: a context-head that indicates the main concept that is held to have an instance within the context, but whose instantiation may flip in a higher context.

3 The Graphical Knowledge Representation

Following these motivations we implement a semantic parser that rewrites a given sentence to a layered semantic graph. The implementation of the parser is done in Java. The semantic graph 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. As will be shown, this approach increases the depth of expressivity and precision because we can, if needed, ignore some sub-graphs and lose precision but we will not lose accuracy. Each semantic graph is a rooted, node-labeled, edge-labeled and directed graph that consists of a dependencies sub-graph, a conceptual sub-graph, a contextual sub-graph, a properties sub-graph and a lexical sub-graph. It can include further sub-graphs as well, such as the co-reference and the temporal sub-graphs. In the following we describe the five obligatory sub-graphs of the sentence *The boy faked the illness.* and what rewritings are required to obtain those graphs.

3.1 The Dependency Graph

The dependency graph represents the full parse of the sentence as this is produced by the Universal Dependencies (UDs). For GKR we use the Stanford CoreNLP Software to produce the dependencies and precisely to produce the enhanced++ UDs (Schuster and Manning, 2016). The enhanced++ UDs make implicit relations between content words more explicit by adding certain relations, e.g. in the case of subjects of control verbs the relation between the subject of the main verb and the control verb is marked by adding an extra edge pointing from the control verb to the subject. The enhanced++ UDs offer a very good basis for our approach because they already deal with many of the phenomena that any semantic parser needs to deal with. The output graph of the Stanford parser is rewritten to our own implementation of the dependency graph (see Figure 2) so that it conforms to the constraints of our layered semantic graph.

3.2 The Conceptual Graph

The conceptual graph shown in Figure 3 (left) contains the basic predicate-argument structure of the sentence as we can extract it from the UDs: *fake* has *boy* as one of its arguments (this is the agent,

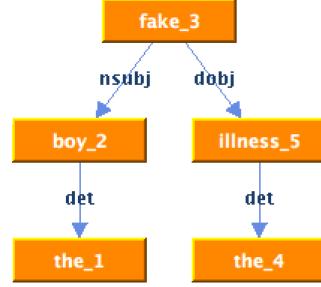


Figure 2: The dependency graph of *The boy faked the illness.*

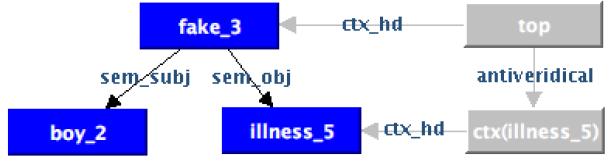


Figure 3: The conceptual graph (left) and the contextual graph (right) of *The boy faked the illness.*

the A0, the semantic-subject or whatever else any other theory might call it) and *illness* as its other argument (again, this is the patient, A1, semantic-object). The conceptual graph is the core of the semantic graph and glues all other sub-graphs together. Thus, if we just look at the concept graph, we know the subject matter of the sentence. A more formal representation might look like this: $\text{fake}(f) \wedge \text{boy}(b) \wedge \text{illness}(i) \wedge \text{agent}(f,b) \wedge \text{patient}(f,i)$. As with AKR (section 2), the variables f , b , and i are not individuals but concepts. The formula $\text{illness}(i)$ does not say that i is an instance of *illness*, but that i is some sub-concept of the lexical concept *illness*. This means that the conceptual graph does not convey all information conveyed by the sentence; it makes no claims about the existence or otherwise of boys or illnesses. But insofar as it goes, the conceptual graph is accurate; what it expresses is correct but incomplete. It allows judgments to be made about semantic similarity between sentences, but not on its own judgments about truth or entailment. The separation of completeness from correctness, and similarity from entailment, is hard to achieve for more conventional logical representations that quantify over individuals.

3.3 The Contextual Graph

The contextual graph provides the existential commitments of the sentence. It introduces a top con-

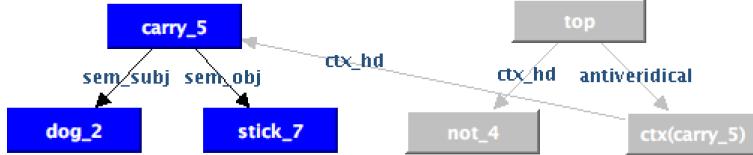


Figure 4: The conceptual graph (left) and the contextual graph (right) of *The dog is not carrying the stick*.

text (or possible world) which represents whatever the author of the sentence takes the described world to be like; in other words, whatever he/she commits to be the “true” world. Below the top context additional contexts are introduced, corresponding to any alternative worlds introduced in the sentence. Each of these embedded contexts makes commitments about its own state of affairs, principally by claiming, through the `ctx_hd` link, that the context’s head concept is instantiated within that context.

Linguistic phenomena that introduce alternative worlds and thus such embedded contexts are negation, disjunction, modals, clausal contexts of belief and knowledge, implicatives and factives, imperatives, questions, conditionals, and distributivity. Apart from the latter four, the rest of the phenomena have already been implemented for this first version of the system by rewriting them to the corresponding contexts. The implicatives and factives are the only contexts that cannot be recognized and dealt with from the surface form of the sentence because their factuality predictions are inherent in their meaning. Therefore, their signatures have to be looked up. For this purpose we use the open source, extended lexicon of Stanovsky et al. (2017) which is based on the works of Karttunen (1971), Karttunen (2012) and Lotan et al. (2013). The lexicon holds more than 2,400 unique words, each assigned to a signature for positive and negative contexts. Predicates are assigned to signatures based on their finite and infinite complements. The extracted signatures are utilized for introducing the necessary contexts.

Our example sentence *The boy faked the illness*. contains such an implicative context. In its contextual graph in Figure 3 (right), the top context says that there is an instance of *faking* in which an instance of a *boy* is faking an instance of an *illness*. The top context has an edge linking it to its head *fake*, which shows that there is an instance of *faking* in this top context. The top context has a second, anti-veridical edge linking it to the con-

text *ctx(illness)* which has *illness* as its head. This head edge asserts that there is an instance of *illness* in this contrary-to-fact context *ctx(illness)*. But since *ctx(illness)* and *top* are linked with an anti-veridical edge, it means that there is no instance of *illness* in the *top* world which is accurate as the *illness* was faked.² Any other concepts, e.g. *boy*, involved in the sentence but not explicitly represented in the contexts graph are taken to exist in the *top* context.

The introduction of contexts or possible worlds to deal with intensional predicates is familiar, though maybe not so much so when combined with reference to concepts rather than individuals. The treatment of Boolean operations like negation and disjunction through contexts is less familiar (though a feature too of AKR). Negation introduces an anti-veridical context. For the sentence *The dog is not carrying the stick*. (see Figure 4) the negated context has as its head the concept of *carrying*, restricted to be a carrying of a stick by the dog. In the negated context, it is asserted that there is an instance of this kind of *carrying*; but in the top context this concept is asserted to be uninstantiated. The impact of the negation is only seen in the context graph; the concept graph is identical for the negated and un-negated sentence. At the moment, we do not deal with morphological negation, e.g. *The boy is unhappy.*, i.e. no additional context is introduced for such negations. Such negations are dealt as normal lexical items for the moment; the mapping to the lexical resources is to account for the correct negative meaning.

Disjunction and conjunction do have an impact on the concept graph. Both introduce an additional complex concept that is the combination of the individual disjoined/conjoined concepts. Each component concept is marked in the concept graph as being an element of the complex concept (Fig-

²Note that definiteness does not project up through presuppositions in a way that predicts existence. Definiteness indicates that some specific kind of *illness* is presupposed, e.g. a (claimed) sore throat that kept the boy away from school, but not some specific individual.

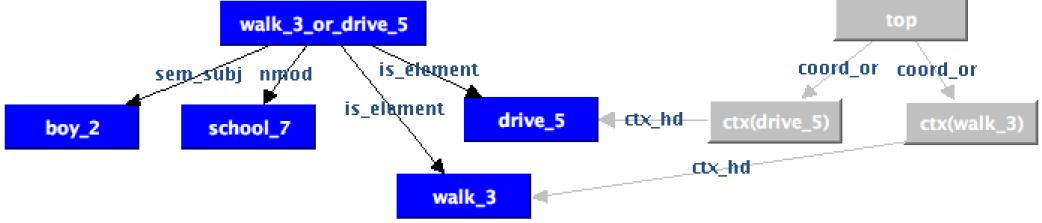


Figure 5: The conceptual graph (left) and the contextual graph (right) of *The boy walked or drove to school.*

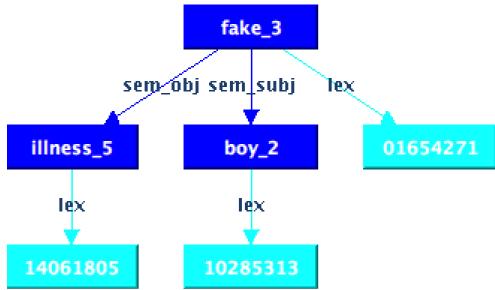


Figure 6: The lexical graph (on top of the conceptual graph) of *The boy faked the illness.*

ure 5, left). The difference between conjunction and disjunction is that disjunction introduces additional contexts for the components of the complex concept (Figure 5, right). These contexts say that in one arm of the disjunct the *walking* concept is instantiated, while in the other arm it is the *driving* concept that is instantiated. The conjunction would just say that both concepts are instantiated in the upper context.

3.4 The Properties Graph

The properties graph (Figure 7) imposes further, mostly non-lexical, restrictions on the graph. It associates the conceptual graph with morphological and syntactical features such as the cardinality of nouns, verbal tense and aspect, finiteness of specifiers, etc. For now, for building the property graph we use our own shallow morphological analysis that is based on the Part-Of-Speech (POS) tags provided by the parser. It is clear that such an analysis cannot capture all complex nuances of phenomena like that of tense and aspect and that it only offers a simplification of those. Still, the properties graph remains accurate; it does not convey all that is there but whatever is conveyed is correct. We plan to implement a temporal graph which is expected to account for the current simplification.

3.5 The Lexical Graph

The lexical graph of Figure 6 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 JIGSAW³ by Basile et al. (2007), WordNet⁴ by Fellbaum (1998) and SUMO⁵ by Niles and Pease (2001) and Pease (2011). For building the lexical graph, the whole sentence is first run through the knowledge-based JIGSAW algorithm which disambiguates each word of the sentence by assigning it the sense with the highest probability. Briefly, JIGSAW exploits the WordNet senses and uses a different disambiguation strategy for each part of speech, taking into account the context of each word. It scores each WordNet sense of the word based on its probability to be correct in that context. The sense with the highest score is chosen as the disambiguated sense and is added as a new node to the lexical graph, with an edge linking the word to its sense. Although the sense is the only lexical information that is visible on the graph, there is more information encoded behind this sense node. Firstly, we encode the SUMO concept corresponding to the disambiguated sense. SUMO is the largest, publicly available ontology that maps WordNet senses to concepts (Niles and Pease, 2003). We access our local copy of the SUMO ontology and extract the concept mapped to the disambiguated sense as well as the hypernyms and hyponyms corresponding to that sense and concept. This information is then stored within the node so that it is easily accessible at all times. The lexical graph can and will be expanded with more information like the one coming from word embeddings. We plan to integrate this component at the next stage of our work.

³Available under <https://github.com/pippokill/JIGSAW>

⁴Available under <http://wordnet.princeton.edu/>

⁵Available under <http://www.ontologyportal.org>

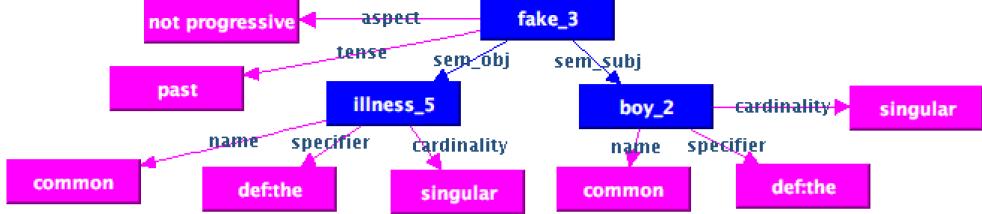


Figure 7: The property graph (on top of the conceptual graph) of *The boy faked the illness*.

4 Evaluation of GKR

4.1 Intrinsic Evaluation

We would like to evaluate our semantic parser to see how many phenomena can already be accurately represented and what should still be improved or implemented. To this end, we use the HP test suite by Flickinger et al. (1987), an extensive test suite with various kinds of syntactic and semantic phenomena, originally created for the evaluation of parsers and other NLP systems. The test suite features 1250 sentences dealing with some 290 distinct syntactic and semantic phenomena and sub-phenomena. Some of the contained sentences are ungrammatical on purpose (and marked as such). For our testing we chose to use a subset of the test suite consisting of 781 sentences (and 180 phenomena, an average of 4.3 sentences pro phenomenon). We decided to exclude ungrammatical sentences (314) and sentences with typos (20) since our testing is aiming at testing the coverage of the semantic graphs and not the accuracy of the parser — which we inevitably and indirectly do as will be shown shortly. We also excluded all sentences (135) with conditionals, anaphora and ellipsis phenomena because such cases are still under implementation and thus yet not part of our system. The test set does not include challenging lexical semantics phenomena, e.g. polysemous words, as it aims at the coverage of syntactic and deeper semantic phenomena. We run the test set of 781 sentences through our semantic parser and got human-readable representations of the semantic graphs which 2 annotators manually evaluated for their correctness. A representation was judged *correct* when the concepts, contexts and properties sub-graphs exactly capture the information they should. If the dependency graph is wrong, then the whole representation is labelled as *parser_error*. Erroneous syntactic parsing will always produce erroneous conceptual and contextual graphs, which we do not deal with at

the moment. The lexical sub-graph was also not judged for the correctness of the selected senses as this would result in evaluating the disambiguation algorithm and the coverage of the lexical resources themselves, which is not the goal of this work. However, any failures in the lexical resources and thus in the lexical sub-graph do not have an impact on the rest of the graphs, which again confirms the flexibility of the layered graph approach. The results of the manual evaluation are shown in Table 1.

Label	Sentences	Percentage
correct	591	75.6%
false	5	0.6%
parser_error	185	23.6%
Total	781	

Table 1: Evaluation results.

Table 1 shows that 185 cases could not be correctly parsed by the Stanford Parser and thus the output semantic representation is inevitably wrong as well. From the remaining 596 sentences for which a correct parse was given, 591 were rewritten to correct semantic graphs and 5 had semantic graphs with missing or wrong information. The overall performance of the system can be seen in Table 2. The initial version of our semantic parser achieves an F-score of 85% when tested on this subset of the HP test suite. Although this test suite and evaluation are not exhaustive, the performance of the system delivers promising results. Note that the relative quality of the integrated tools, e.g. the syntactic parser, the implicatives-factives lexicon,

Metric	Percentage
Precision	0.99
Recall	0.76
F-score	0.85

Table 2: Overall performance of the system.

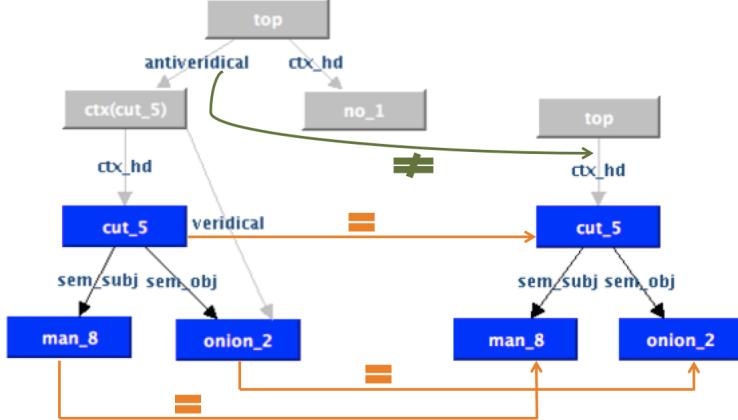


Figure 8: Schematic NLI computation for the pair A = *No onion is being cut by a man.* (left) B = *An onion is being cut by a man.* (right).

etc., has a direct impact on the overall quality of the semantic representations and the performance of our parser.

4.2 Schematic Computation of Natural Language Inference

We would like to very briefly demonstrate how GKR facilitates semantic processing tasks, such as natural language inference (NLI) and semantic similarity, by describing the inference computation of the pair A = *No onion is being cut by a man.*, B = *An onion is being cut by a man.*⁶ For doing NLI (see Figure 8) we determine specificity relations⁷ between pairs of individual concept nodes, one from the premise (A) and one more from the hypothesis (B) sentence. In the figure these correspond to equality relations and are represented by the orange arrows. These initial specificity judgments can then be updated with any further restrictions placed on the nodes from the properties and lexical graphs. The context graph is then used to determine which concepts are instantiated or uninstantiated within which contexts. In our example, we can see that *cut* is instantiated, i.e. is the *ctx_head* of the top of B but is antiveridical in the top of A. Similarly, in B *onion* is veridical in top (and therefore it is not explicitly represented) while in A it is veridical only in context of *cut* and since *ctx(cut)* is antiveridical in top, *onion* is also antiveridical in top through transitivity. As a final step for inference, instantiation and specificity are

combined to determine entailment relations.

In the same process, if we choose to ignore the context graphs and the instantiation of concepts, we can also measure semantic similarity — which does not require judgments about truth or entailment. The semantic similarity between the two sentences can be measured on the basis of the concepts graphs of the sentences. Since the concept graph represents “what is talked about”, the comparison of the concepts graphs can compute the overall similarity by computing the similarity of the different concept pairs of the two sentences and merging them together.

5 Future Work

At this point, old-school semanticists will probably be asking: but what about quantifier scope? This is a rarer phenomenon than the literature would have you believe. The primary reading for a sentence like *Three boys ate five pizzas* involves no scope variation: there were just three boys and five pizzas, and eating. This cumulative reading is difficult to express in standard logical representations without recourse to branching quantifiers, or to treating three and five not as generalized quantifiers but as cardinality restrictions on existential quantifiers. It is an inelegance that scoped readings are the default in these representations, while being the exception in practice.

That being said, quantifier scope — or rather, distributivity — does occur; *take two tablets three times* really does involve six tablets. We regard distributivity as context inducing (Figure 9). The distributional context has two arcs into the concept graph. In addition to the normal context head arc,

⁶The pair comes from the SICK corpus (Marelli et al., 2014).

⁷The specificity relations are taken as discussed in MacCartney and Manning (2007) and Crouch and King (2007).

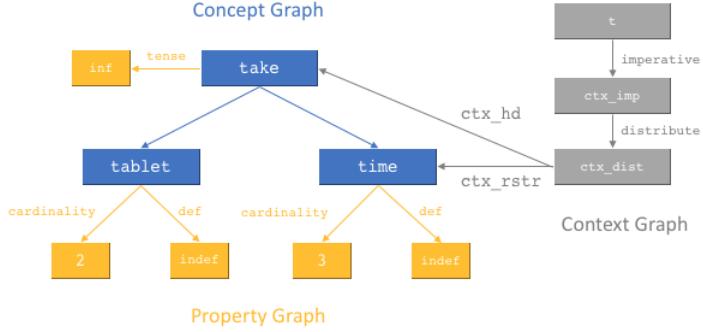


Figure 9: Distributivity for *Take two tablets three times.*

which marks the body of the distribution, there is a context restriction arc that marks the concept to be distributed over: in this case the *times* that comprise individual sub-concepts of the concept *3-times*; see (van den Berg et al., 2001) for more details on individual sub-concepts. For each individual sub-concept in the distributive restriction, there is asserted to be an instance of the head concept further restricted by the individual sub-concept.

Distributive contexts are similar to our proposed conditional contexts, which also have head (consequent) and restriction (antecedent) arcs. This is reminiscent of the use of conditionals to express universal quantification in Discourse Representation Theory (Kamp and Reyle, 1993). That quantification is treated as having a modal aspect should not be that surprising. In first order modal logic, modal operators switch the context of evaluation of sub-formulas by altering the assignment of a possible world. Quantifiers switch the context of evaluation by altering the assignment to a variable. Both, in other words, switch contexts of evaluation. Our contextual treatment of distributivity just makes this similarity more apparent.

The proposed layered semantic graph can involve further sub-graphs as mentioned before. One of them may be the co-reference sub-graph which should link together any elements referring to the same entities, e.g. to resolve any pronouns involved or to identify two elements as “identical”, i.e. as referring to the same entity. A simple example of those kinds of linking can be seen in Figure 10 for the sentence *John, our neighbor, loves his wife.* Here, the pronoun *his* is resolved to its referent *John* and *John* is set as “identical” to *neighbor*. Similar co-reference graphs expanding over the level of a single sentence should be able to account for some inter-sentential semantics where the co-referring entities of different sentences, e.g.

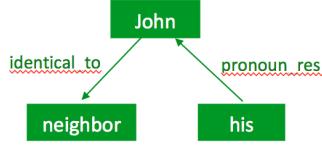


Figure 10: Co-reference graph for *John, our neighbor, loves his wife.*

of the premise and of the hypothesis in the natural language inference task, are inter-connected to each other and thus facilitate the further processing.

6 Related Work

How does GKR differ from its precursor, AKR? While the two representations are very close, they differ in that a) AKR is based on the syntax produced by LFG while GKR is based on UD and that b) AKR is rather flat-structured while GKR is based on graphs. Although LFG is probably more informative and could offer us for free some of the features that we need to implement extra for UD, its parsing is either not robust enough or not openly available in comparison to state-of-the-art dependency parsers. Also, it is not straightforwardly combinable with other state-of-the-art techniques that we wish to utilize, e.g. with word embeddings. Additionally, a graph-based representation is beneficial for our purposes, as already discussed in Section 1. Last but not least, AKR and its most recent revision in Boston et al. (forthcoming) is proprietary software and our intention is to produce a semantic parser that can be offered freely and openly to the community.

A more recent meaning representation is the AMR (Banarescu et al., 2013), which aims at introducing a semantic representation language with which a given sentence can be translated to its se-

mantic formula. The representation is based on manual annotation of the structures and is thus expensive, while the attempts for automatic creation of AMRs are currently showing low accuracy (Flanigan et al., 2014; Wang et al., 2015). But this is not the only drawback: AMR ignores function words, tense, articles and prepositions which means that important information for the semantic processing remains unused. Additionally, AMR has limited expressive power for universal quantification (Bos, 2016), models negation in an inconvenient way (Bos, 2016) and does not make a distinction between real and *irrealis* events (as in our example *The boy faked the illness.*). Another disadvantage is the fact that AMR is biased towards English as pointed out by the creators. Although our system is also built for English and the lexical resources necessary are also language-dependent, the approach and GKR itself are highly language-independent. Furthermore, the fact that the sentential representation is conflated in only one graph does not facilitate semantic tasks that require stepwise access to different kinds of information, e.g. semantic similarity tasks.

A more venerable representation is DRT (Kamp and Reyle, 1993). This follows a first-order, individual based approach to predicate-argument structure rather than the concept based approach of AKR. However, the ability to name sub-Discourse Representation Structures (DRSs), and have those sub-DRSs act as arguments of (modal) predicates is very closely connected to our use of contexts. DRT shows a willingness to freely mix individual and context-denoting discourse referents, which tends to bring a highly realist approach to possible worlds in its wake. GKR, on the other hand, is careful to impose a kind of blood-brain barrier between concepts and contexts.

DepLambda (Reddy et al., 2016) uses a lambda calculus based method to transform dependencies into logical forms. Similar to GKR in availing itself of general dependency parsers, the semantic representation is essentially non-graphical, and we are unsure about how existential commitments are dealt with and whether this approach could really be practically used for the tasks of inference and reasoning. We are also skeptical about the fact that the semantic representations of semantically-identical sentences, e.g. a passive/active sentence, do not look alike, as the authors themselves observe.

Although AKR, AMR, DRT and DepLambda are the closest to our representations, there are a couple of other approaches that can be viewed as a step towards producing semantic representations for semantic processing. Firstly, there is the work of Schuster and Manning (2016) who bring UDs a step further by enhancing them with more explicit relations which are needed for any kind of further semantic processing. Their work is the basis of GKR, not only because the produced UDs are of high quality (Schuster and Manning, 2016), but also because different linguistic phenomena that can change how a semantic representation looks like are already solved, e.g. the subject of raising verbs is made explicit. There are still cases that are not optimally solved, e.g. copulas and expletives, and we hope that they can be improved in the future. A similar attempt is the system PropS by (Stanovsky et al., 2016) which is designed to explicitly express the proposition structure of a sentence. The system abstracts away from the syntactic structure by adding relations such as *outcome* and *condition* for conditionals while not becoming too abstract as AMR is. It is thus going this “next” step towards semantics without however offering a more complete semantic structure.

7 Conclusions

We have presented an expressive, graph-based semantic formalism that supports semantic parsing, as well as modal and hypothetical textual inference. Future work will account for the formal definitions of the notions presented in this paper. The first version of the parser is publicly available under https://github.com/kkalouli/GKR_semantic_parser. A companion paper (Crouch and Kalouli, 2018) discusses in more detail the benefits of such layered graphs for semantic representation.

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 Linguistic Annotation Workshop*.
- 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.
- Emily M. Bender, Dan Flickinger, Stephan Oepen, Woodley Packard, and Ann Copestake. 2015. *Layers of Interpretation: On Grammar and Compositionality*. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 239–249, London, UK. Association for Computational Linguistics.
- J. Berant, A. Chou, R. Frostig, and P. Liang. 2013. Semantic Parsing on Freebase from Question-Answer Pairs. In *Empirical Methods in Natural Language Processing (EMNLP)*.
- Martin van den Berg, Cleo Condoravdi, and Richard Crouch. 2001. Counting concepts. In *Proceedings of the Thirteenth Amsterdam Colloquium*, pages 67–72.
- Danny G. Bobrow, Bob Cheslow, Cleo Condoravdi, Lauri Karttunen, Tracy Holloway-King, Rowan Nairn, Valeria de Paiva, Charlotte Price, and Annie Zaenen. 2007a. PARC’s Bridge question answering system. In *Proceedings of the GEAF (Grammar Engineering Across Frameworks) 2007 Workshop*.
- Danny G. Bobrow, Cleo Condoravdi, Richard Crouch, Valeria de Paiva, Lauri Karttunen, Tracy Holloway-King, Rowan Nairn, Charlotte Price, and Annie Zaenen. 2007b. *Precision-focused Textual Inference*. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, RTE ’07, pages 16–21, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Johan Bos. 2016. Expressive Power of Abstract Meaning Representations. *Computational Linguistics*, 42(3):527–535.
- Marisa Boston, Richard Crouch, Erdem Özcan, and Peter Stuble. forthcoming. Natural language inference using an ontology. In Cleo Condoravdi, editor, *Lauri Karttunen Festschrift*.
- Danqi Chen and Christopher D. Manning. 2014. A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of EMNLP 2014*.
- Cleo Condoravdi, Richard Crouch, John Everett, Valeria De Paiva, Reinhard Stolle, Danny Bobrow, and Martin Van Den Berg. 2002. Preventing Existence.
- Richard Crouch and Aikaterini-Lida Kalouli. 2018. Named Graphs for Semantic Representations. In *Proceedings of *SEM 2018, to appear*.
- Richard Crouch and Tracy Holloway King. 2007. *Systems and methods for detecting entailment and contradiction*. US Patent 7,313,515.
- C. Fellbaum. 1998. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.
- Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of ACL*.
- D. Flickinger, J. Nerbonne, I.A. Sag, and T. Wasow. 1987. Toward Evaluation of NLP Systems. Hewlett-Packard Laboratories. In *24th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer, Dordrecht.
- Lauri Karttunen. 1971. Implicative verbs. *Language*, 47:340–358.
- Lauri Karttunen. 2012. Simple and phrasal implicatives. In *Proceedings of *SEM 2012*, pages 124–131.
- Amnon Lotan, Asher Stern, and Ido Dagan. 2013. Truthteller: Annotating Predicate Truth. In *Proceedings of NAACL-HLT 2013*, page 752–757.
- Bill MacCartney and Christopher D. Manning. 2007. Natural logic for textual inference. In *Proceedings of the ACL Workshop on Textual Entailment and Paraphrasing*.
- 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*.
- John Maxwell and Ron Kaplan. 1996. An Efficient Parser for LFG. In *Proceedings of the First LFG Conference*.
- Rowan Nairn, Cleo Condoravdi, and Lauri Karttunen. 2006. Computing relative polarity for textual inference. *Inference in Computational Semantics (ICoS-5)*, pages 20–41.
- 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.
- Ian Niles and Adam Pease. 2003. Linking Lexicons and Ontologies: Mapping Wordnet to the Suggested Upper Merged Ontology. In *Proceedings of the IEEE International Conference on Information and Knowledge Engineering*, pages 412–416.
- Valeria de Paiva, Danny G. Bobrow, Cleo Condoravdi, Richard Crouch, Ron Kaplan, Lauri Karttunen, Tracy Holloway-King, Rowan Nairn, and Annie Zaenen. 2007. Textual inference logic: Take two. In *Proceedings of the Workshop on Contexts and Ontologies: Representation and Reasoning, Workshop associated with the 6th International Conference on Modeling and Using Context*.

Adam Pease. 2011. *Ontology: A Practical Guide*. Articulate Software Press, Angwin, CA.

Vittorio Perera, Tagyoung Chung, Thomas Kollar, and Emma Strubell. 2018. Multi-Task Learning for parsing the Alexa Meaning Representation Language. In *Proc AAAI*.

Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. 2016. Transforming Dependency Structures to Logical Forms for Semantic Parsing. *Transactions of the Association for Computational Linguistics*, 4:127–140.

Sebastian Schuster and Christopher D. Manning. 2016. Enhanced English Universal Dependencies: An Improved Representation for Natural Language Understanding Tasks. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*.

Ed Stabler. 2017. Reforming AMR. In *Formal Grammar 2017. Lecture Notes in Computer Science*, volume 10686. Springer.

Gabriel Stanovsky, Judith Eckle-Kohler, Yevgeniy Puzikov, Ido Dagan, and Iryna Gurevych. 2017. Integrating Deep Linguistic Features in Factuality Prediction over Unified Datasets. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 352–357.

Gabriel Stanovsky, Jessica Ficler, Ido Dagan, and Yoan Goldberg. 2016. Getting More Out Of Syntax with Props. *CoRR*, abs/1603.01648.

Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015. A transition-based algorithm for AMR parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.