Knowledge Integration for Commonsense Reasoning with Default Logic

Priit Järv[1][a], Tanel Tammet[1][b], Martin Verrev[1][c] and Dirk Draheim[2][d]
[1]*Applied Artificial Intelligence Group, Tallinn University of Technology, Tallinn, Estonia*
[2]*Information Systems Group, Tallinn University of Technology, Tallinn, Estonia*

Keywords: Commonsense Reasoning, Default Logic, Knowledge Base.

Abstract: Commonsense reasoning in artificial intelligence is the problem of inferring decisions and answers regarding mundane situations. Several research groups have built large knowledge graphs with the goal of capturing some aspects of commonsense knowledge. Using these knowledge graphs for problem solving and question answering is a subject of active research. Our contribution is encoding and integrating knowledge graphs like Quasimodo, ConceptNet, and Wordnet for symbolic reasoning. A major challenge in symbolic commonsense reasoning is coping with contradictory and uncertain knowledge, which we handle by extending first order logic with numeric confidences and default logic rules. To our knowledge this is the first large scale commonsense knowledge base seriously using default logic. We give several examples of the proposed representation and solving questions on the basis of the knowledge base built.

1 INTRODUCTION

Knowledge representation for commonsense reasoning is a non-trivial problem. (McCarthy, 1989) remarks that a knowledge representation language that can be used to reason with generalized concepts is "ambitious", while (Davis, 2017) has a more direct opinion that "we do not, by any means, know how to represent all or most of the commonsense knowledge needed" in such reasoning tasks. The view in the natural language processing community is far more optimistic, e.g. (Trinh and Le, 2018) suggesting that a deep neural model has good understanding of context and common sense. More recent quantitative results of language models on difficult problems appear to support that claim. However, skepticism has been expressed due to well known "statistical black box" properties of numerical models. (He et al., 2021) measure whether neural language models understand the logical structure in CSR problems, concluding that their "reasoning ability could have been overestimated".

There is an emerging view that the synthesis of different approaches in AI – logical, probabilistic and machine learning – is required for common sense (Marcus, 2020), resulting in recent popularity of neurosymbolic reasoning. Typically the approach is to add symbolic reasoning capability to the neural model, see, for example, (Garnelo and Shanahan, 2019; Riegel et al., 2020; Arabshahi et al., 2021). We approach the same goal of synthesis from the opposite direction: supplementing symbolic reasoning with mechanisms for plausible reasoning as described in (Davis, 2017).

Our goal is to develop a general question answering system. We have set a target of being able to reason at the level of a small child. This includes tasks like causal and spatial reasoning, counting and manipulating sets. By incrementally adding knowledge in the form of common sense facts and inference rules and identifying further gaps, we can empirically find whether the target is realistic and what is the scope of the knowledge base required in terms of size and areas covered.

We are developing both a default logic reasoner and a knowledge base. This paper focuses only on the latter and presents the automated construction of the knowledge base and the representation of uncertain and contradictory knowledge using default logic.

[a] https://orcid.org/0000-0001-7725-543X
[b] https://orcid.org/0000-0003-4414-3874
[c] https://orcid.org/0000-0003-4890-9283
[d] https://orcid.org/0000-0003-3376-7489

Järv, P., Tammet, T., Verrev, M. and Draheim, D.
Knowledge Integration for Commonsense Reasoning with Default Logic.
DOI: 10.5220/0011532200003335
In *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2022) - Volume 2: KEOD*, pages 148-155
ISBN: 978-989-758-614-9; ISSN: 2184-3228

2 PLAUSIBLE REASONING

Our knowledge representation extends first order logic (FOL) with numeric confidences and non-monotonic rules with exceptions to enable plausible reasoning. We will assume that the FOL statements are converted to a conjunctive normal form.

We use a representation language that consists of a limited set of predicates, describing common relations between entities, classes, properties and events. The set of relations is similar to what is used in ConceptNet (Speer et al., 2017).

We assign a numeric confidence to each clause in the range $0 \dots 1$. Each derived clause will have a confidence value based on the confidence of clauses used in the derivation. Literals in a clause do not contain confidence information: the latter is present only at the level of clauses.

Example 1 (Confidence of clauses)**.** *We want to express that Garfield refers to a cat with a 0.9 confidence and that cats are tabby, with 0.7 confidence. Additionally, there is already a weak indication that garfield is tabby:*

IsA(cat, X) → Property(tabby, X) : 0.7
IsA(cat, garfield) : 0.9
Property(tabby, garfield) : 0.1

We have built a system for plausible reasoning with default logic rules. We give a brief overview of the reasoner below. A more complete description can be found in (Tammet et al., 2022).

Our reasoner will multiply confidences while using modus ponens and cumulate the confidence of the result with the given confidence 0.1, resulting with *Property(tabby, garfield) : 0.667*

Inconsistencies in the knowledge base are handled by (a) requiring that each derivation of an answer contains clauses stemming from the question posed, (b) performing searches both for the question and its negation and returning the resulting confidence calculated as a difference of the confidences found by these two searches.

We use the *default logic* of Reiter (Reiter, 1980) for encoding rules with exceptions. Default logic extends classical logic with default rules of the form $\alpha(x) : \beta_1(x),...\beta_n(x) \vdash \gamma(x)$ where a *precondition* α(x), *justifications* $\beta_1(x),...\beta_n(x)$ and a *consequent* $\gamma(x)$ are first order predicate calculus formulas whose free variables are among $x = x_1,...,x_m$. For every tuple of individuals $t = t_1,...,t_n$, if the precondition $\alpha(t)$ is derivable and none of the *negated* justifications $\neg\beta(t)$ are derivable, then the consequent $\gamma(t)$ can be derived. Without losing generality we assume that $\alpha(x)$ and $\gamma(x)$ are clauses and $\beta_i(x)$ are positive or negative atoms.

We encode a default rule as a clause by concatenating into one clause the precondition and consequent clauses and blocker atoms $\$block(p_1, \neg\beta_1), ..., \$block(p_n, \neg\beta_n)$ built from the justifications.

Example 2 (Default logic rule)**.** *The "birds can fly" default rule is represented as a clause*

$$\neg Bird(X) \lor Fly(X) \lor \$block(0, \neg(Fly(X)))$$

where X is a variable and $\neg(Fly(X))$ *encodes the negated justification. The first argument of the blocker (0 above) encodes priority information. Blocker atoms collect the substitutions applied during the derivation of clauses.*

Our approach to handling default rules is to delay justification checking of blocker atoms until a first-order proof is found and then perform recursively deepening checks with diminishing time limits. If a clause that contains only blocker atoms is derived during the proof search, the proof is found, as without the blockers it would be an empty clause.

Our system first produces a potentially large number of different candidate proofs and then enters a recursive checking phase. When checking a blocker with a given priority, it is not allowed to use default rules with a lower priority. The main assumed use of priorities is preferring rules associated with more specific concepts to these of more general concepts, as proposed in (Brewka, 1994). To this end we have built a data structure and an efficient algorithm for checking whether an English word w_s occurs below a more general word w_g on a branch of Wordnet.

3 KNOWLEDGE GRAPHS

Quasimodo and ConceptNet are large knowledge graphs with different goals and representation principles. The main focus of our paper is on converting them to a FOL representation augmented with default rules and confidences, so that a question answering or problem solving reasoner can use them together in a single derivation.

The converted data sets are represented in JSON using the specification proposed in (Tammet and Sutcliffe, 2021; Tammet, 2020). In this paper we will use conventional FOL syntax for examples.

The commonsense knowledge base construction consists of

1. Converting source relations or sentences to FOL rules using a limited set of relations.

2. Assigning confidences to rules.
3. Adding blocker atoms to default rules.
4. Normalizing terms coming from heterogeneous data sources.
5. Finding semantic similarities.
6. Adding general inference rules.

The resulting knowledge base consists of rules that come directly from data sources (Sections 3.1–3.3) and the added layer of similarity and general inference rules (Sections 3.4–3.5). The current knowledge graphs are limited in terms of diversity and coverage of commonsense knowledge, so we are looking to grow our knowledge base using more sources – ATOMIC$^{20}_{20}$, large neural network language models and raw text (Section 3.6).

3.1 Quasimodo

Quasimodo (Romero et al., 2019) gathers candidate assertions from search engine auto-completion suggestions and question answering forums like Reddit, Quora etc. These are further corroborated using other sources with the help of a learned regression model. A ranking step adds typicality and plausibility.

The latest version 4.3 of Quasimodo has a size of roughly one gigabyte and contains over 6 million rows of triples augmented with negative/positive sign s, salience score σ, source information etc (Table 1). Triple elements are snippets of natural language texts, not formal or standardized predicate/argument structures. If $s = 1$ then the relation is negative. The salience score σ indicates whether the given property is typically associated with the given subject. Some of the salience scores do not match our common knowledge well.

We convert Quasimodo to logic by mapping each relation to a predicate. We generate three types of rules: class membership, known relations and generic subject-verb-object relations.

Class membership rules cover both the taxonomy of classes and instance membership. Quasimodo does not contain many instances: most of its knowledge concerns classes. In this paper, we use *IsA* for class membership rules. **Known relations** are mapped to a small set of predicates with a known meaning, for example $Location$ and $HasA$, the latter meaning possession of something. The Quasimodo relations ”live”, ”be in” and ”be on” are mapped to the $Location$ predicate. The advantage of using known relations is that we can create general inference rules for them (see Section 3.5).

All other relations are represented as a generic subject-verb-object (SVO) relation. For example, in Table 1, ”fly” does not have a matching predicate in our representation language and the row will be encoded as an SVO. Reasoning with the SVO predicates is still possible, but requires that the vocabulary used in terms is limited or allows fuzzy matching.

If the Quasimodo relation is negative, then we negate the conclusion of the rule. The plausibility score is taken directly ”as is” to be the confidence of the rule.

We treat everything except taxonomy rules as default rules. We use a taxonomy graph describing the hierarchy of classes. If the subject of the rule is is present in our taxonomy graph, we create a blocker atom for this rule.

The **taxonomy graph** is derived from the hypernym and hyponym relations of Wordnet. We create a directed graph and remove all edges which are one-directional, e.g. where word w_i is a hypernym of w_j but w_j is not a hyponym of w_i. This is sufficient to make an acyclic graph. We delete all nodes that have no remaining edges. We then assign the topological sort order of the graph by the hyponym relation as the identifier of the word. When making a blocker atom, we assign the identifier of the word (synset) matching the subject of the rule from the taxonomy as the blocker priority number.

If word w_g, with the identifier g is more general than word w_s, then $g < s$. This is used as a heuristic during the proof search. If $g > s$ then w_g is not more general. Otherwise, a short search on the taxonomy graph by the the hypernym relation is needed to confirm that w_g is more general.

We describe how we parse longer text fragments from Quasimodo in Section 3.3.

Example 3 (Logical rules from Quasimodo)**.** *A taxonomy relation:*

$$IsA(bird, penguin) : 0.96$$

A known relation, also present in Table 1. The number 84487 is the identifier of ”penguin” from the taxonomy graph. This rule means that we have 0.99 confidence that penguins do not fly, unless there is some more specific penguin that has this capability:

$$IsA(penguin, X) \rightarrow (\neg Capability(fly, X) \lor \\ \$block(84487, Capability(fly, X))) : 0.99$$

An generic subject-verb-object relation ”eat” with no blocker. This rule contains a compound noun ”leopard seal” encoded as one term:

$$IsA(leopard\_seal, X) \rightarrow SVO(penguin, eat, X) : 0.92$$

3.2 ConceptNet

ConceptNet (Speer et al., 2017) is another triple-based knowledge graph. The English subset consists

Table 1: Excerpt of Quasimodo data.

Subject	Relation	Object	s	σ
bird	can	flying	1	0.99
bird	fly	over the acropolis	1	0.99
birds eye	be protected	from winds when flying	0	0.38
penguin	can	fly	1	0.99
penguin	can	fly	0	0.42
penguin	has_property	unable to fly	0	0.76

of ca 21 million edges. It is built from the earlier Open Mind Common Sense project by adding data obtained Wiktionary, WordNet, DBPedia, crowd-sourced human input and several other sources. The edges (triples) use a pre-determined set of 36 relations. The majority of triples use a vague *RelatedTo* relation, followed by mostly linguistic *FormOf*, *DerivedForm*, *IsA* and *Synonym*. The percentage of more specific relations is relatively low. In comparison, Quasimodo contains more detailed knowledge, although it also contains more erroneous facts and requires further natural language processing for effective use. The ConceptNet relations also have a weight parameter, which appears to be not well maintained. There are no negative facts.

As an example, ConceptNet contains nine *IsA* relations for penguin (animal, bird, seabird, weapon etc) and just three additional non-linguistic relations – *AtLocation* (zoo and Antarctica) and *Desires* (enough to eat).

We convert ConceptNet using the same approach as Quasimodo, using edge weights as rule confidences. The main difference is that ConceptNet already has a limited set of relations which are easy to map to predicates.

3.3 Text Fragments

All parts of Quasimodo triples may contain text fragments that require natural language parsing to interpret properly. Given the Quasimodo triple *(mutual friend, show up, on facebook)*, we could encode it as $SVO(on\_facebook, show\_up, mutual\_friend)$. This representation is poor, because we can not infer anything about "friend" or "Facebook", which the triple clearly is about. The words "show", "up", "on", currently split between multiple elements, may be better represented as a generic relation of one object being contained by another. Our goal is to encode the triple as:

$$IsA(friend, X) \wedge Property(mutual, X) \rightarrow Location(facebook, X)$$

This represents the structural transformation. We do not propose any particular mapping to predicates, e.g. the predicates *Property* and *Location* may change in the course of our ongoing work. For brevity, we omitted the confidence and the blocker clause.

The existing solutions to extract relations from text fall under three categories: triple extraction (Angeli et al., 2015; Gashteovski et al., 2017; West et al., 2021), parsing from a controlled language (Fuchs et al., 2008) and full first order logic extraction (Basile et al., 2016). There are tool kits that do not offer a ready-made solution but can be used to write an application to do the required translation, e.g. DeepDive (Zhang, 2015). The triple extraction tools like OpenIE (Angeli et al., 2015) are not practical here, because their output is similar to Quasimodo triples, requiring still further parsing. Using a controlled language would merely shift the problem to translating from natural language to controlled language. That leaves the solutions that directly output first order logic.

The gold standard in translating raw text to full first order logic is Boxer (Bos, 2008), more recently packaged into a full text to meaning representation pipeline KNEWS (Basile et al., 2016). The Boxer translation of the example triple is

$$\begin{aligned}&\exists A,B,C,D,E\\ &\quad a1up(A) \wedge r1Manner(C,A) \wedge r1on(C,B) \wedge\\ &\quad n1facebook(B) \wedge r1Actor(C,E) \wedge v1show(C) \wedge\\ &\quad n1friend(E) \wedge a1mutual(D) \wedge r1Theme(D,E)\end{aligned}$$

The output is essentially a graph like a syntactic parse tree. The edges represent semantic rather than syntactic relations. The structure is still different from our desired representation and further translation would be needed. The Boxer pipeline processed 1000 rules/s on our test system which is fast enough for converting large triple sets and 3 times faster than our own pipeline.

Because the output of existing semantic parsing solutions requires further, potentially non-trivial translation, we have opted to implement the semantic interpretation ourselves based on an out-of-the box syntactic parser. About 5 million relation triple elements, or 27% of Quasimodo are text fragments consisting of multiple words. The grammatical structures

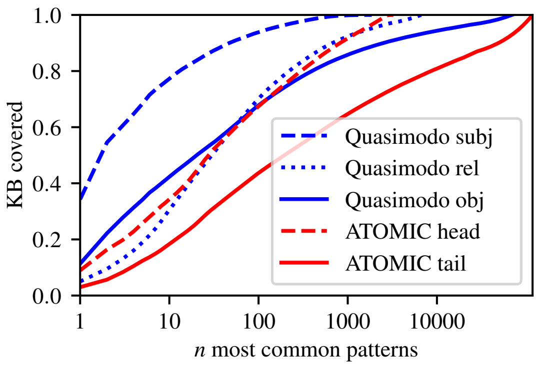

Figure 1: Cumulative distribution of grammatical structures in knowledge graphs. The vertical axis is the fraction of the triples covered by the top n most common structures.

Table 2: Most common grammatical structures of the object in Quasimodo as constituency parse trees. w_i are placeholders for words.

Constituency tree	Count
(NP (JJ w_1) (NN w_2))	147574
(NP (NN w_1) (NN w_2))	141173
(NP (JJ w_1) (NNS w_2))	72167
(NP (NN w_1) (NNS w_2))	47692

in these fragments have long-tailed distributions (Figure 1), where a small number of common patterns cover the majority of fragments. For example, there are a total of 69178 distinct structures in the object element, but the most common 100 structures cover approximately 60% of the cases. The most common structures are noun phrases (Table 2). The object and subject elements share the same common structures, but for the relation the vast majority of structures are clauses containing a verb phrase.

The distribution of structures allows us to interpret a large fraction of the fragments by recognizing relatively few distinct patterns. Our fragment parser tokenizes and tags the input using the Stanza NLP pipeline (Qi et al., 2020). We pre-process the resulting consituency parse trees by simplifying them and detecting compound nouns. If an n-gram of tokens matches a dictionary word from Wiktionary (Ylonen, 2022), or if Stanza's NER recognizes a sequence of tokens as a named entity, the tokens will be concatenated into a single noun chunk. We then match the parse tree to grammatical structure patterns. For example, if the object of a triple matches the pattern *(NP (JJ w_1) (NN w_2))* then the object will be w_2, with the added property or qualifier w_1.

Currently we have implemented 19 patterns for subjects and objects and 9 patterns for relations. Our parser matched 82% of multi-word subjects and objects and 40% of multi-word relations in Quasimodo. The coverage can be developed further by adding patterns, but with diminishing returns.

Example 4 (Fragment parsing)**.** *The negative Quasimodo fact "Norhern Ireland,have,rugby team" is encoded as:*

$$\begin{aligned} IsA(northern\_ireland, X) \rightarrow \neg HasA(sk(X), X) \wedge \\ IsA(team, sk(X)) \wedge \\ Property(rugby, sk(X)) \end{aligned}$$

"Northern Ireland" is a named entity, encoded as one term. The fragment "rugby team" is split by the parser. The rule is skolemized because new variables introduced to the right hand side must be existentially bound to a left hand side variable.

3.4 Term Normalization and Similarity

The objects, properties and other elements of the knowledge base we reason about can have multiple textual representations. If some facts are about *bird* and others about *birds*, for the inference engine these are disjoint pieces of knowledge. To effectively integrate multiple knowledge bases, we need to inform the reasoner that both *bird* and *birds* refer to the same class. Because we use a limited set of predicates, in our knowledge base the multiple representations occur as terms. We normalize the terms, choosing a single representation for what we consider the same concept. We use English language lemmatization and reduction to Wordnet synsets using Stanza (Qi et al., 2020) and nltk (Bird et al., 2009).

Some concepts are different but similar. For example, we know that *dog* and *wolf* are the same in many ways. This can be useful for extending our reasoning capabilities – if we are missing the fact that wolves have fur, we may infer than from similarity to dogs. There are also terms that we should normalize, but cannot because they are not present in Wordnet. Indirect semantic similarity helps recognizing such synonyms.

Similarity-based inferences cannot be statically resolved before the proof search. Consider the following knowledge:

$$IsA(south\_pole, X) \rightarrow Property(cold, X) : 0.9$$
$$IsA(south\_pole, X) \rightarrow Property(dry\_land, X) : 0.9$$
$$IsA(north\_pole, X) \rightarrow \neg Property(dry\_land, X) : 0.98$$

Knowing that North Pole and South Pole are similar, it is correct to infer that North Pole is cold. It is not correct to infer that there is dry land at North Pole. This can only be discovered during the proof search, particularly because the negative *Property* fact may itself be a product of proof search and not be statically present.

We generate facts $Similar(term_j, term_i) : s_{ij}$ where $term_j$ are the terms in k-neighborhood of $term_i$ and s_{ij} is the semantic similarity. We include terms that appear in subject and object. We exclude multi-word text fragments which we failed to parse correctly and where similarities to other terms may be coincidental. Each term is assigned a semantic vector computed with SpaCy's[1] "en_core_web_lg" model. Because neighborhood queries at this scale (10000-500000 items) and dimensionality (300) are expensive even with the specialized Ball tree index, we use a fast approximate index[2].

Semantic similarity is not clearly defined – the distance functions have mathematical definitions but they operate on vectors which are highly subjective and method-dependent. We make the following adjustments to achieve subjectively better "common sense" similarity with SpaCy's general purpose vectors.

Neighborhood Filtering. Semantic vectors do not always follow human common sense. With the "en_core_web_lg" model, the similarity of *Northern Europe* and *Southern Europe* is 0.97, but the similarity of *Italy* and *Southern Europe* is 0.66. The terms *northern* and *southern* themselves are semantically very similar, but in human usage as identities, they often convey a cultural and geographical distance. Similar effects happen with antonyms (*warm*, *cold*), proper nouns, enumerators (*one*, *two*, *Monday*, *Tuesday*). We apply a soft filter which weakens the similarity between a pair of terms if we detect that the terms contain antonyms, or if enumerators or proper nouns are present in both terms.

Frequent Terms. Out of 308322 ConceptNet and Quasimodo terms we selected for similarity data, 61% appear only once.

With the size of neighborhood $k = 5$, the neighborhoods of common (more than 10 occurrences) terms consist of 48% rare (1 occurrence) terms. So, rare terms prevent seeing similarities between more common terms. By definition, rare terms only cover a small fraction of the knowledge base. To facilitate connecting more facts by similarity, we include only common terms for generating similarity facts. Intuitively, the resulting similarities contain fewer surprises.

[1] https://github.com/explosion/spaCy

[2] https://github.com/spotify/annoy

3.5 Inference Rules

The knowledge graphs Quasimodo and ConceptNet contain factual statements like "parrots are birds" and "birds can fly", but to infer that "parrots can fly" from the knowledge, general inference rules are needed. Our knowledge base construction includes creating the rules for transitivity of the *IsA* relation and symmetry of *Similar*. Generalization rules (Example 5) are created for predicates that are inferred from classes. Generalization is drawing a general conclusion from specific evidence. Observing that a person can drive a car allows us to generalize that the person can drive other cars as well. This kind of inference is not logically sound but follows "common sense". The opposite direction of inference, from general to narrow or specific, is supported by transitivity of *IsA*.

Example 5 (Generalization rule)**.** *The following relations can be obtained from Quasimodo:*

$$IsA(teenage\_mutant\_ninja\_turtles, X) \rightarrow SVO(pizza, eat, X) : 0.85$$
$$IsA(junk\_food, pizza) : 0.02$$

With the automatically generated rule

$$IsA(W,X) \wedge SVO(X,Y,Z) \rightarrow SVO(W,Y,Z) : 0.8$$

we can infer that teenage mutant ninja turtles eat junk food, although the confidence is < 0.02

Predicates describing classes can also be inferred by class similarity. We automatically generate similarity rules from a list of known predicates, for example:

$$Similar(X,Y) \wedge Property(Z,Y) \rightarrow Property(Z,X) : 0.7$$

With our knowledge encoding method, rules apply to instances and subclasses of a class. Reasoning about classes should be done by populating or instantiating the classes:

Example 6 (Class instantiation)**.** *Given the clause:*

$$IsA(cat, X) \rightarrow Property(tabby, X) : 0.7$$

A FOL reasoner, perhaps unintuitively for a human, cannot answer the query $\neg Property(tabby, cat)$*. However, we can populate the class with an instance:* $IsA(cat, cat\_instance)$ *and then query about the instance:* $\neg Property(tabby, cat\_instance)$.

3.6 Other Data Sources

Our ongoing focus is to add more data sources to diversify and extend our default logic knowledge base.

Table 3: Excerpt of ATOMIC$^{20}_{20}$ data. X and Y are variables.

LHS	Relation	RHS
X votes for Y	xIntent	to give support
X votes for Y	oReact	grateful
bread	ObjectUse	make french toast

The ATOMIC$^{20}_{20}$ knowledge graph (Hwang et al., 2021) contains over a million everyday inferential knowledge tuples about entities and events. The bulk of knowledge encodes human attitudes, wanting, needs, effect of action etc: commonsense information about social interactions (Table 3).

Part of ATOMIC$^{20}_{20}$ relations are ConceptNet-like, e.g. *AtLocation* and *CapableOf*. These relations can be treated as subject-relation-object triples and converted similarly to the Quasimodo data, augmented with fragment parsing (Section 3.3). The rest of the data describe relations of events to other events or objects and social interactions. The grammatical structure in these relations is more diverse than Quasimodo (Figure 1), so pattern matching may be less effective. We intend to use event-based encoding with thematic roles, proposed in many earlier papers, e.g. (Furbach and Schon, 2016), but with the goal of having a minimal rather than rich set of thematic roles.

(West et al., 2021) demonstrated that high quality knowledge graphs can be extracted from large language models. The machine generated ATOMIC^{10x} is somewhat larger than ATOMIC$^{20}_{20}$ and in the same format. The source code of their experiment is available so there is potential to generate new relation triples tailored for specific purposes.

As shown with Quasimodo, our knowledge base construction pipeline can convert triples consisting of English words or short simple text fragments. Therefore, a suitable text source like Simple English Wikipedia can be exploited by parsing it with OpenIE (Angeli et al., 2015).

4 RESULTS

We have made the default logic knowledge bases we created available[3] along with the reasoning system GK, a number of examples and a tutorial.

We experimented with a knowledge base translated from Quasimodo and English language ConceptNet, consisting of 1786884 facts, 24 inference rules and 68886 similarity rules. Our reasoning system answers trivial questions quickly. For example, GK proves in 4 seconds that birds fly and in 2 seconds that penguins cannot fly, both with confidence 1. The latter proof successfully invalidates the contradicting flying proof stemming from a taxonomy fact in Quasimodo by proving that the blocker holds. To the question whether babies have hair, the system properly gives neither a positive nor a negative confirmation: Quasimodo contains both a positive and a negative fact for this statement, both with a high confidence. These examples show that our automatically constructed knowledge base can be successfully used to reason with contradictory knowledge.

However, when we try to use Quasimodo and ConceptNet knowledge for nontrivial questions, we almost always fail. For example, we can infer that sheep have wool, but cannot infer that a sheep with no wool would be cold in cold weather. This is because our knowledge base is missing several facts for that chain of reasoning, starting with the fact that sheep need wool to keep warm. In other words, there are still not enough commonsense rules. To identify and map the knowledge gaps we need quantitative evaluation. We are actively working on parsing natural language questions and small sets of premises, so that we can test our approach on existing commonsense reasoning benchmarks.

[3]http://logictools.org/gk/

5 SUMMARY

This paper describes our knowledge representation for commonsense reasoning with default logic. Our present target is general question answering at the level of the understanding of a small child. We present the methods to convert the Quasimodo and ConceptNet knowledge graphs to our knowledge base. We demonstrated the extraction and use of contradictory and uncertain knowledge in question answering with our inference engine built for plausible reasoning (Tammet et al., 2022). We cannot yet conclusively state whether a default logic knowledge base is a competitive alternative to neural language models, for that quantitative evaluation is needed. Ongoing and future work includes integrating more data sources and developing capability to interpret existing natural language benchmark questions.

REFERENCES

Angeli, G., Premkumar, M. J. J., and Manning, C. D. (2015). Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354.

Arabshahi, F., Lee, J., Bosselut, A., Choi, Y., and Mitchell, T. (2021). Conversational multi-hop reasoning with neural commonsense knowledge and symbolic logic rules. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7404–7418.

Basile, V., Cabrio, E., and Schon, C. (2016). Knews: Using logical and lexical semantics to extract knowledge from natural language. In *Proceedings of the European conference on artificial intelligence (ECAI) 2016 conference*.

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

Bos, J. (2008). Wide-coverage semantic analysis with boxer. In Bos, J. and Delmonte, R., editors, *Semantics in Text Processing. STEP 2008 Conference Proceedings, Venice, Italy, September 22-24, 2008*. Association for Computational Linguistics.

Brewka, G. (1994). Adding priorities and specificity to default logic. In *European Workshop on Logics in Artificial Intelligence*, pages 247–260. Springer.

Davis, E. (2017). Logical formalizations of commonsense reasoning: a survey. *Journal of Artificial Intelligence Research*, 59:651–723.

Fuchs, N. E., Kaljurand, K., and Kuhn, T. (2008). Attempto controlled english for knowledge representation. In Baroglio, C., Bonatti, P. A., Maluszynski, J., Marchiori, M., Polleres, A., and Schaffert, S., editors, *Reasoning Web, 4th International Summer School 2008, Venice, Italy, September 7-11, 2008, Tutorial Lectures*, volume 5224 of *Lecture Notes in Computer Science*, pages 104–124. Springer.

Furbach, U. and Schon, C. (2016). Commonsense reasoning meets theorem proving. In *German Conference on Multiagent System Technologies*, pages 3–17. Springer.

Garnelo, M. and Shanahan, M. (2019). Reconciling deep learning with symbolic artificial intelligence: representing objects and relations. *Current Opinion in Behavioral Sciences*, 29:17–23.

Gashteovski, K., Gemulla, R., and Del Corro, L. (2017). Minie: Minimizing facts in open information extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2630–2640.

He, W., Huang, C., Liu, Y., and Zhu, X. (2021). Winologic: A zero-shot logic-based diagnostic dataset for winograd schema challenge. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3779–3789.

Hwang, J. D., Bhagavatula, C., Le Bras, R., Da, J., Sakaguchi, K., Bosselut, A., and Choi, Y. (2021). (comet-) atomic 2020: On symbolic and neural commonsense knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6384–6392.

Marcus, G. (2020). The next decade in AI: four steps towards robust artificial intelligence. *CoRR abs/2002.06177*.

McCarthy, J. (1989). Artificial intelligence, logic and formalizing common sense. In *Philosophical logic and artificial intelligence*, pages 161–190. Springer.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., and Manning, C. D. (2020). Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Reiter, R. (1980). A logic for default reasoning. *Artificial Intelligence*, 13(1–2):81–132.

Riegel, R., Gray, A., Luus, F., Khan, N., Makondo, N., Akhalwaya, I. Y., Qian, H., Fagin, R., Barahona, F., Sharma, U., et al. (2020). Logical neural networks. *arXiv preprint arXiv:2006.13155*.

Romero, J., Razniewski, S., Pal, K., Pan, J. Z., Sakhadeo, A., and Weikum, G. (2019). Commonsense properties from query logs and question answering forums. In Zhu, W., Tao, D., Cheng, X., Cui, P., Rundensteiner, E. A., Carmel, D., He, Q., and Yu, J. X., editors, *Proc. of CIKM'19 – the 28th ACM Intl. Conf. on Information and Knowledge Management*, pages 1411–1420. ACM.

Speer, R., Chin, J., and Havasi, C. (2017). ConceptNet 5.5: An open multilingual graph of general knowledge. In Singh, S. P. and Markovitch, S., editors, *Proc. of AAAI'2017 – the 31st AAAI Conf. on Artificial Intelligence*, pages 4444–4451. AAAI.

Tammet, T. (2020). JSON-LD-LOGIC homepage. https://github.com/tammet/json-ld-logic.

Tammet, T., Draheim, D., and Järv, P. (2022). GK: implementing full first order default logic for commonsense reasoning (system description). In Blanchette, J., Kovács, L., and Pattinson, D., editors, *Automated Reasoning - 11th International Joint Conference, IJCAR 2022, Haifa, Israel, August 8-10, 2022, Proceedings*, volume 13385 of *Lecture Notes in Computer Science*, pages 300–309. Springer.

Tammet, T. and Sutcliffe, G. (2021). Combining jsonld with first order logic. In *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*, pages 256–261. IEEE.

Trinh, T. H. and Le, Q. V. (2018). A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*.

West, P., Bhagavatula, C., Hessel, J., Hwang, J. D., Jiang, L., Bras, R. L., Lu, X., Welleck, S., and Choi, Y. (2021). Symbolic knowledge distillation: from general language models to commonsense models. *CoRR*, abs/2110.07178.

Ylonen, T. (2022). English machine-readable dictionary. https://kaikki.org/dictionary/English/index.html.

Zhang, C. (2015). *DeepDive: a data management system for automatic knowledge base construction*. PhD thesis, The University of Wisconsin-Madison.