Applications of Ontologies (TAO)

Leveraging Ontologies and Machine Learning to Address Challenges in Handling Transfer-Type Arithmetic Word Problems

Suresh Kumar
(CS18D007)

Advisor: Prof P Sreenivasa Kumar
Artificial Intelligence & Databases (AIDB) Lab
Computer Science & Engineering
Indian Institute of Technology Madras

Outline

- Arithmetic Word Problems (AWPs)
- Challenging Tasks Related to AWPs
- SOTA - Limitations - Research Gap
- Background (RDF, OWL, SWRL)
- Applications / Proposed Approaches
- Conclusions

Arithmetic Word Problems

- Arithmetic Word Problems (AWPs)
  - Numerical problems expressed in natural language (English)
  - Popular AWP datasets are All-Arith[1], MAWPS[2], SVAMP[3], Dolphin[4], etc.

1. Roy, S. and Roth, D. Unit dependency graph and its application to AWP solving, AAAI 17
2. Koncel-Kedziorski et al. MAWPS: A math word problem repository, ACL 16
3. Patel et al. Are NLP Models really able to solve simple Math Word Problems?, NAACL 21
4. Huang et al. How well do computers solve math word problems? large-scale dataset construction and evaluation, ACL 16

Arithmetic Word Problems

- **Some Examples**
  - **P1:** Mike had 9 dimes in his bank. Sara had 12 dimes. Sara borrowed 5 dimes from Mike. How many dimes does Sara have now?
  - **P2:** John has 12 yellow balloons. Mike has 14 yellow balloons. Mike gave 9 yellow balloons to John. How many yellow balloons does John have now?
  - **P3:** John is 5 years older than Sara. In 3 years, the sum of their ages will be 35 years. How old are John and Sara now?
  - **P4:** Sara is twice as old as her brother, Mike. Five years ago, the sum of their ages was 20. How old are Sara and Mike now?

Arithmetic Word Problems

- **Some Examples**
  - **P1:** Mike had 9 dimes in his bank. Sara had 12 dimes. Sara borrowed 5 dimes from Mike. How many dimes does Sara have now? — **Transfer-Type**
  - **P2:** John has 12 yellow balloons. Mike has 14 yellow balloons. Mike gave 9 yellow balloons to John. How many yellow balloons does John have now? — **Transfer-Type**
  - **P3:** John is 5 years older than Sara. In 3 years, the sum of their ages will be 35 years. How old are John and Sara now? — **Age-WP**
  - **P4:** Sara is twice as old as her brother, Mike. Five years ago, the sum of their ages was 20. How old are Sara and Mike now? — **Age-WP**

Challenging Tasks Related to AWPs

- Automatic Solving
- Generation
- Validity Checking
- Repairing

Challenging Tasks Related to AWPs

- Automatic Solving $(x + y, x - y)$
- Generation (Ag1 - 9 units, Ag2 - 15 units, Transfer ?)
- Validity Checking
- Repairing

State-of-the-Art Approaches

			Popular SOTA Approaches
1. For Solving →	1	**non-LLM models** (Rule based, DL based, etc)	TRNN[1] Text2Math[2]
	2	**LLM models** (General, Specialized)	chatGPT[3] Bard/Gemini[4] MAmmoTH[5]
2. For Generation →	1	**Logic Programming-based**	Theme-rewriting[6]
	2	**DL-based**	MAGNET[7]
3. For Validity Checking →	1	**DL-based**	Controllable Generation[8]

1. Wang et al., Template based MWP solvers with RNNs, AAAI 2019
2. Zou et al., Text2Math: End-to-End Parsing Text into Math Expressions, EMNLP 2019
5. Yue et al., MAmmoTH: Building Math Generalistic Models, Arxiv 2024
6. Koncel-Kedziorski et al., A theme re-writing approach for generating…, EMNLP 2016
7. Zhou et al., Towards Generating MWPs from Equations and Topics, INLG 2019
8. Wang et al., MWP Generation with Mathematical Consistency and…, EMNLP 2021

3. https://chat.openai.com/
4. https://gemini.google.com/

Limitations of the existing Approaches (1/2)

For Solving

- **Are NLP Models really able to Solve Simple Math Word Problems ? (Patel et al., 2021, ACL)**
  - Test the SOTA systems on BOW representations of AWPs and AWPs ***without*** the question part of the problem
    - Rely on simple heuristics in the training instances to make their predictions
    - Capability to solve simple AWPs is overestimated

- **Why are NLP Models Fumbling at Elementary Math? A Survey of Deep Learning based Word Problem Solvers (Sundaram et al., Arxiv 2022)**
  - Analyze the existing AWP solver systems
    - With a small word change
    - Minor change in the mathematical structure
    - Do not adequately model both language and math

Limitations of the existing Approaches (1/2)

For Solving

- **Are NLP Models really able to Solve Simple Math Word Problems ? (Patel et al., 2021, ACL)**
  - Test the SOTA systems on BOW representations of AWPs and AWPs ***without*** the question part of the problem
    - Rely on simple heuristics in the training instances to make their predictions
    - Capability to solve simple AWPs is overestimated

- **Why are NLP Models Fumbling at Elementary Math? A Survey of Deep Learning based Word Problem Solvers (Sundaram et al., Arxiv 2022)**
  - Analyze the existing AWP solver systems
    - With a small word change
    - Minor change in the mathematical structure
    - Do not adequately model both language and math

Sequential reasoning ?

Limitations of the existing Approaches (2/2)

For Generation

- **Are NLP Models really able to Solve Simple Math Word Problems ? (Patel et al., 2021, ACL)**
  - Proposed SVAMP dataset
    - Size: 1000 word problems
    - Approx 130 TC-AWPs, about 40% are invalid/language-issue

- **MWP Generation with Mathematical Consistency and Problem Text Constraints (Wang et al, 2021, EMNLP)**
  - Generated problems - lack mathematical validity

Limitations of the existing Approaches (2/2)

For Generation

- **Are NLP Models really able to Solve Simple Math Word Problems ? (Patel et al., 2021, ACL)**
  - Proposed SVAMP dataset
    - Size: 1000 word problems
    - Approx 130 TC-AWPs, about 40% are invalid/language-issue
- **MWP Generation with Mathematical Consistency and Problem Text Constraints (Wang et al, 2021, EMNLP)**
  - Generated problems - lack mathematical validity

How to ensure mathematical validity? **Can we repair ?**

Research Gap

- Learning-based approaches
  - non-LLM models - mimic the existing patterns in the training set
  - LLMs - exhibit significant reasoning deficiency[1]
    - repeat conceptual mistakes[1]
- Only statistical findings are sufficient - best handled by ML/DL
- Reasoning portion - can we rely on only statistical findings?
- Need for semantically rich models (incorporate domain knowledge)
- Encoded domain knowledge is not sharable

1. Bubeck et al., Sparks of Artificial General Intelligence: Early experiments with GPT-4, Microsoft Research, 2023

Applications

- Solving simple TC word problems
- Validity checking and repairing of machine-generated examples
- Generating complex examples and how to solve them

What's Required...

Mike has 9 books. Sara has 15 books. Mike gave 3 books to Sara.
How many books does Sara have now ?

What's Required...

Mike has 9 books. Sara has 15 books. Mike gave 3 books to Sara.
How many books does Sara have now ?

How to develop an ontology-based solver ?

What's Required...

Mike has 9 books. Sara has 15 books. Mike gave 3 books to Sara.
How many books does Sara have now ?

1. **TBox : concepts - properties - axioms**

What's Required...

Mike has 9 books. Sara has 15 books. Mike gave 3 books to Sara.
How many books does Sara have now ?

1. **TBox : concepts - properties - axioms**
2. **ABox - concrete facts**

What's Required...

Mike has 9 books. Sara has 15 books. Mike gave 3 books to Sara.
How many books does Sara have now ?

1. **TBox : concepts - properties - axioms**
2. **ABox - concrete facts**
3. **SWRL rules**

What's Required...

Mike has 9 books. Sara has 15 books. Mike gave 3 books to Sara.
How many books does Sara have now ?

Concepts ? **Properties ?**

What's Required...

Mike has 9 books. Sara has 15 books. Mike gave 3 books to Sara.
How many books does Sara have now ?

↓

S1 - Mike has 9 books.
S2 - Sara has 15 books.
S3 - Mike gave 3 books to Sara.
S4 - How many books does Sara have now ?

Concepts ? **Properties ?**

What's Required...

Mike has 9 books. Sara has 15 books. Mike gave 3 books to Sara.
How many books does Sara have now ?

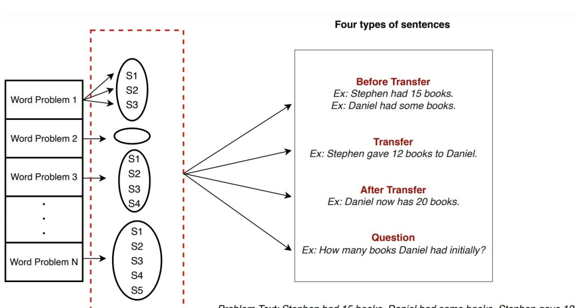
↓

S1 - Mike has 9 books.
S2 - Sara has 15 books.
S3 - Mike gave 3 books to Sara.
S4 - How many books does Sara have now ?

Concepts ? **Properties ?**

Agent, Quantity Agent *hasQaunt* Quantity

Types of sentences in TC-AWPs



Problem Text: Stephen had 15 books. Daniel had some books. Stephen gave 12 books to Daniel. Daniel now has 20 books. How many books Daniel had initially?

TC Ontology

Important Concepts

1. AWP
2. Before Transfer (BT), Transfer (TR), After Transfer (AT), Question (QS)
3. Person, Agent
4. Positive Quantity, Negative Quantity
5. TC-Quantity, TR-Quantity, Non-TR-Quantity
6. Subtrahend-Quantity, Minuend-Quantity
7. ValidBT, ValidTR, ValidAT, ValidQS
8. ValidAWP, InvalidAWP

TC Ontology

Important Properties

1. hasBT, hasTR, hasAT, hasQS
2. involvesAgent
3. hasToAgent, hasFromAgent
4. hasQuant, isOwnedBy
5. hasTRQuantity
6. hasLost, isLostBy
7. hasGained, isGainedBy
8. quantValue, quantType
9. hasSequenceNumber, hasTRSequence
10. asksObjType

Object Properties: 1-7
Data Properties 8-10

TC Ontology

Important Axioms

1. $\text{Agent} \equiv \text{Person} \sqcap \exists \text{hasQuant.TCQuantity}$
2. $\text{TRQuantity} \sqsubseteq \text{TCQuantity}$
3. $\text{TRQuantity} \sqcap \text{NonTRQuantity} \sqsubseteq \bot$
4. $\text{hasGained} \equiv \text{isGainedBy}^{-}$, $\text{hasLost} \equiv \text{isLostBy}^{-}$
5. $\text{MQ} \equiv \text{NonTRQuantity} \sqcap \exists \text{isOwnedBy.Agent}$
6. $\text{SQ} \equiv \text{TRQuantity} \sqcap \exists \text{isGainedBy.Agent} \sqcap \exists \text{isLostBy.Agent}$

coming next...

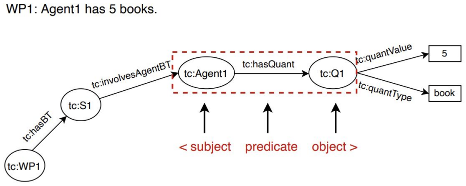
- Background - RDF, RDFS, Ontology
- How domain ontology plays an important role

Background - RDF and RDFS

- Resource Description Framework (RDF)
  - A W3C standard for describing resources and data exchange
  - Captures information in the form of triples
  - RDF triple : < subject, predicate, object >
  - RDF uses Uniform Resource Identifiers (URIs)
    - To uniquely identify all the entities of a triple
- Resource Description Framework-Schema (RDFS)
  - A vocabulary description language that provides a way of giving limited ontological information
  - Domain vocabulary :
    - concepts/classes, properties/roles, individuals/instances
  - Limitation: expressivity

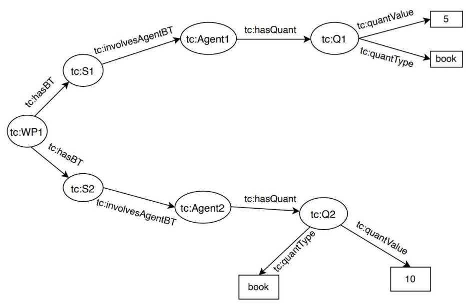
RDF representation of an example word problem

WP1: Agent1 has 5 books.



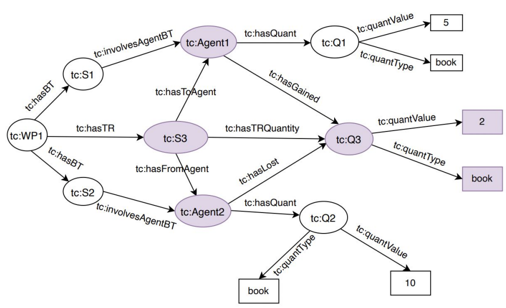
RDF representation of an example word problem

WP1: Agent1 has 5 books. Agent2 has 10 books.



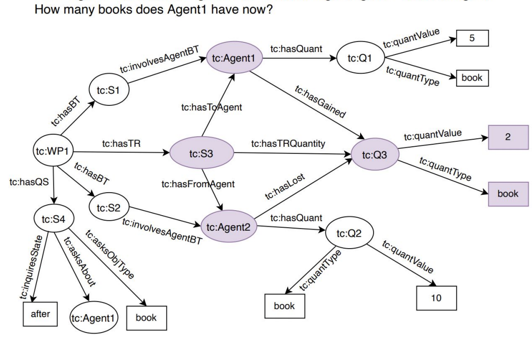
RDF representation of an example word problem

WP1: Agent1 has 5 books. Agent2 has 10 books. Agent2 gave 2 books to Agent1.



RDF representation of an example word problem

WP1: Agent1 has 5 books. Agent2 has 10 books. Agent2 gave 2 books to Agent1.
How many books does Agent1 have now?



Background - Ontology

OWL and SWRL

- Web Ontology Language (OWL)
  - A formal knowledge modeling framework
  - Family of languages designed to represent rich and complex knowledge
  - Has more constructs than RDFS, more expressive
  - Enables efficient automated reasoning
  - Based on Description Logic (decidable subset of FOL)
  - Modeling Languages
    - Domain description - RDFS
    - Logic - OWL and SWRL
- Semantic Web Rule Language (SWRL)
  - Rule language for Semantic Web
  - Addresses some of the OWL's limitations (constructs meant for properties)
  - Includes built-ins (e.g. swrlb:equal, swrlb:notEqual, swrlb:lessThan, etc.)
  - hasParent(?x1, ?x2) ^ Man(?x2) ➔ hasFather(?x1, ?x2)

The role of domain ontology in proposed works - Brief

- Offers Conceptualization
  - Helps facilitate better problem understanding
- Knowledge captured helps create a mental model of the TC domain
- Knowledge captured helps
  - Infer quantities involved in the math operations (solving)
  - In validity checking
  - Generate word problems that involve multiple object transfers
- Helps repair the word problems (examples with minor generation error)
- Knowledge triples are reused during the generation task (simple ➔ complex)

Important pieces of knowledge captured by Ontology (1/2)

- Concepts - capture all resources that conceptually belong together
  - AWP
  - BT, ValidBT
  - TR, ValidTR
  - Agent
  - TCQuantity
- Properties - capture relationships
  - Who owns what (Agent hasQuant TCQuantity)
  - Agent lost a quantity (Agent hasLost TRQuantity)
  - Agent gained a quantity (Agent hasGained TRQuantity)
  - hasFromAgent, hasToAgent, hasTRQuantity

Important pieces of knowledge captured by Ontology (2/2)

- Knowledge captured by axioms
  - **A1:** Agent $\equiv$ Person $\sqcap$ $\exists$ hasQuant.TCQuantity
  - **A2:** TRQuantity $\sqsubseteq$ TCQuantity
  - **A3:** ValidBT $\equiv$ =1(involvesAgentBT.(=1hasQuant.BTQuantity))
  - **A4:** ValidTR, ValidAT, ValidQS
  - **A5:** SQ $\equiv$ TRQuantity $\sqcap$ $\exists$ isGainedBy.Agent $\sqcap$ $\exists$ isLostBy.Agent
  - **A6:** hasGained $\equiv$ isGainedBy^{-}, hasLost $\equiv$ isLostBy^{-}
- Knowledge captured by SWRL rules
  - About math operations (equal, greater-than, add/sub)
  - One can not transfer more than what (s)he owns
  - Unit compatibility checks
  - About under-represented cases
  - To perform computation based of the sequence of transfers
  - Helps validate sentence structures, to perform repairs

Ontology for transfer-type word problems

Summary

 # Concepts = 23,

 # Properties = 28,

 # Axioms = 109,

 # Individuals = vary

What's Required…

Mike has 9 books. Sara has 15 books. Mike gave 3 books to Sara.
How many books does Sara have now ?

1. **TBox : concepts - properties - axioms**
2. **ABox - concrete facts**
3. **SWRL rules**

Ontology for transfer-type word problems

A-Box (for an example word problem)

T1

Q1 Q2 Q3

Agent1 has 12 books. Agent2 has 4 books. Agent2 gave 3 books to Agent1.
How many books does Agent1 have now?

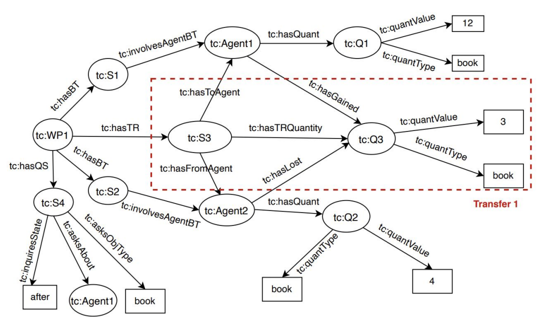
QS

AWP(P1)
Agent(Agent1)
Agent(Agent2)
Transfer(T1)
Question(QS)
hasTR(P1, T1)
hasQS(P1, QS)

hasQuant(Agent1, Q1)
hasQuant(Agent2, Q2)
hasFromAgent(T1, Agent2)
hasToAgent(T1, Agent1)
hasLost(Agent2, Q3)
hasGained(Agent1, Q3)
asksAbout(QS, Agent1)
asksObjType(QS, “book”)

@ to capture the value and type information of Q1, Q2, and Q3 data properties **quantValue** and **quantType** are used

ABox in RDF representation

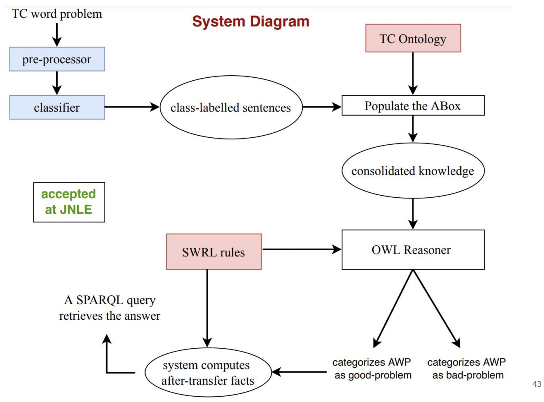


Problem Text: Stephen has 12 books. Daniel has 4 books. Daniel gave 3 books to Stephen. How many books does Stephen have now ?

Applications

- Solving simple TC word problems
- Validity checking and repairing of machine-generated examples
- Generating complex examples and how to solve them

Solving simple TC-AWPs using domain knowledge and learning



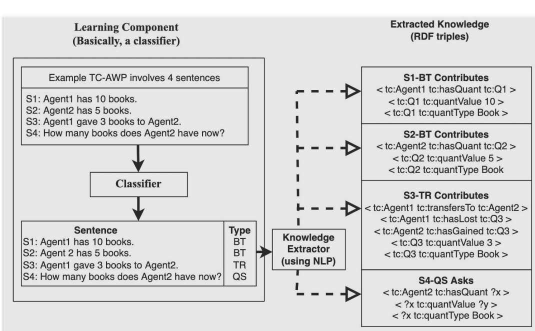
Solving simple TC-AWPs using domain knowledge and learning

Pre-processing

- Some example Transfer Case AWPs
  - Stephen picked 37 oranges. Daniel picked 10 oranges. Daniel gave 4 oranges to Stephen. How many oranges does Stephen have now ?
  - Mike grew 6 carrots. Bob grew 3 carrots. Bob took 3 carrots from Mike. How many carrots does Mike have now ?
  - Joan has 6 notebooks and 2 pens. Sara has 3 notebooks. She gave 2 notebooks to Joan. How many notebooks does Joan have now ?
- Pre-processing : sentence simplification, sentence normalization (NLTK library, POS tag information)
- Pre-processed word problem text
  - Agent1 has 6 notebooks. Agent1 has 2 pens. Agent2 has 3 notebooks. Agent2 gave 2 notebooks to Agent1. How many notebooks does Agent1 have now ?

Solving simple TC-AWPs using domain knowledge and learning

Why sentence-type information is important ?



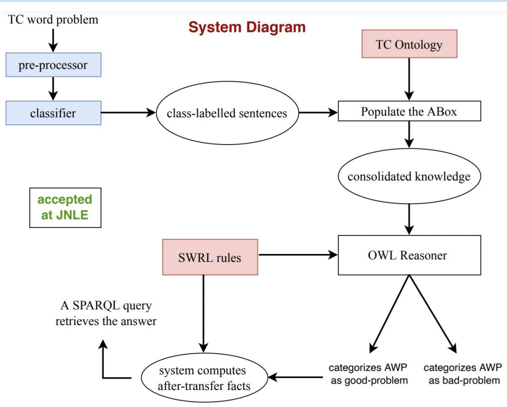
Solving simple TC-AWPs using domain knowledge and learning

Sentence Classifier

- Various classes are:
  - Before-Transfer (BT)
  - Transfer (TR)
  - After-Transfer (AT)
  - Question (QS)
- Features
  - BoW, N-grams, Positional-Weights
  - For example: Positional-Weight (QS) > Positional-Weight (BT)
- Popular Algorithms
  - Naive Bayes
  - Decision Tree
  - Random Forest
  - AdaBoost
  - XGBoost

Solving simple TC-AWPs using domain knowledge and learning

System Diagram



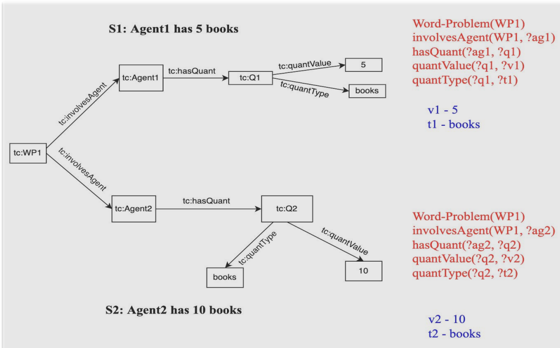
Solving simple TC-AWPs using domain knowledge and learning

Pseudo code of the proposed solver

Input: WP_i, Ontology O
Set $S_i, C_i = \{\}$ ▷ S_i: normal-sentences of WP_i, C_i: class-labeled sentences of WP_i
$S \leftarrow$ Sentence-Splitter(WP_i) ▷ Let S contain k sentences
for *iteration* $j = 0, 1, 2, \ldots, k-1$ **do**
  $C_i \leftarrow$Classifier(S_{ij}) ▷ S_{ij} - the j^{th} sentence in WP_i
end for
$\tilde{O}_i \leftarrow$Pop-Onto(C_i, O) ▷ where $\tilde{O}_i$ contains consolidated knowledge
Synchronize-Reasoner($\tilde{O}_i$)
  • If reasoner determines WP_i is a Bad-Problem
    • System labels WP_i a Bad-Problem and terminates
  • If reasoner determines WP_i is a Good-Problem
    • System labels WP_i a Good-Problem and generates after-state facts(AF_i)
Query $\leftarrow$ Query-Gen(S_q) ▷ S_q is the question sentence from C_i
Answer $\leftarrow$ Result-Gen(Query, AF_i)

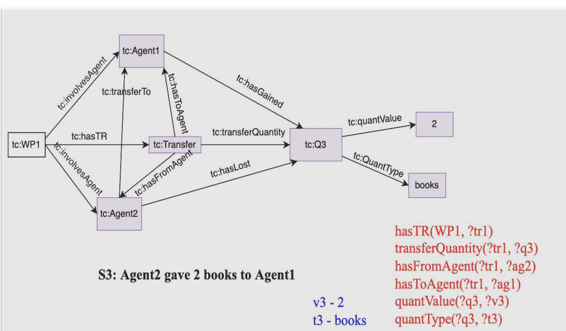
Solving simple TC-AWPs using domain knowledge and learning

How SWRL component affects the RDF graph of an example WP



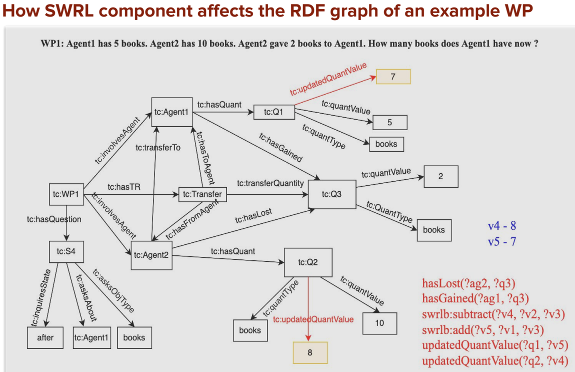
Solving simple TC-AWPs using domain knowledge and learning

How SWRL component affects the RDF graph of an example WP



Solving simple TC-AWPs using domain knowledge and learning

How SWRL component affects the RDF graph of an example WP



Solving simple TC-AWPs using domain knowledge and learning

SWRL rule to compute after-transfer facts

$$a_1 \wedge a_2 \wedge \ldots \wedge a_k \rightarrow a_{k+1} \wedge a_{k+2} \wedge \ldots \wedge a_n$$

C1 ~ Word-Problem(WP1) ^ involvesAgent(WP1, ?ag1) ^ hasQuant(?ag1, ?q1) ^ quantValue(?q1, ?v1) ^ quantType(?q1, ?t1)

C2 ~ Word-Problem(WP1) ^ involvesAgent(WP1, ?ag2) ^ hasQuant(?ag2, ?q2) ^ quantValue(?q2, ?v2) ^ quantType(?q2, ?t2)

C3 ~ hasTR(WP1,?tr1) ^ transferQuantity(?tr1,?q3) ^ hasFromAgent(?tr1,?ag2) ^ hasToAgent(?tr1,?ag1) ^ quantValue(?q3,?v3) ^ quantType(?q3,?t3) ^ hasLost(?ag2,?q3) ^ hasGained(?ag1,?q3) ^ swrlb:equal(?t1,?t2) ^ swrlb:equal(?t2, ?t3) ^ swrlb:subtract(?v4, ?v2, ?v3) ^ swrlb:add(?v5, ?v1, ?v3)

C1 ^ C2 ^ C3 → updatedQuantValue(?q1, ?v5) ^ updatedQuantValue(?q2, ?v4)

Solving simple TC-AWPs using domain knowledge and learning

Results - Sentence Classification Accuracy

Model	Accuracy	Precision	Recall	F1-score
Naive Bayes	0.60	0.44	0.50	0.47
Decision Tree	0.86	0.68	0.71	0.68
Random Forest	0.86	0.68	0.71	0.68
AdaBoost	0.95	0.72	0.75	0.73
XGBoost	0.95	0.72	0.75	0.73

Table 1: Sentence classification results on **test** data

Rep.	Measure	RF	AdaBoost	XGBoost
Basic	Accuracy	0.55	0.74	0.74
Custom	Accuracy	0.86	0.95	0.95

Table 2: Effect of feature engineering

Solving simple TC-AWPs using domain knowledge and learning

Results - Word Problem Solving Accuracy

System	Dataset		% of TR cases		Accuracy	
ALGES	**All-Arith**	*DS*	**38.14**	*≈ 8*	60.4	-
ExpTree	All-Arith	DS	38.14	≈ 8	79.4	26.11
UNITDEP	All-Arith	DS	38.14	≈ 8	81.7	28.78
MathDQN	All-Arith	DS	38.14	≈ 8	72.68	30.06
Text2Math	AI2+IL	DS	42.80	≈ 8	83.2	-
MDK	All-Arith	DS	38.14	≈ 8	73.32	-
T-RNN	MAWPS	DS	27.60	≈ 8	66.8	39.1
Proposed	Arith-Tr	DS(Tr)	100	100	**92**	**65**

- #AWPs- All-Arith: 831, AI2+IL: 957, MAWPS: 2373, Dolphin-S(DS): 1878, DS(Tr): 146, **Arith-Tr**: 794

Table 4: Experimental analysis of proposed system on various datasets. All the results are on % scale.