2023/8/1 (Tue)
ESSLLI2023 : Pros&Comps Workshop

Mizuki Iinuma, Sora Tagami, Yuta Takahashi, and Daisuke Bekki
(Ochanomizu University, Tokyo, Japan)

Toward an inference procedure by type checking algorithm for Neural DTS

1.
Introduction

Natural Language Inference (NLI)

NLI is the task of determining whether a natural-language hypothesis can be inferred from given premises [MacCartney and Manthat, 2008]

Premises		Hypothesis
Every noodle is cheap Pasta is noodle	$\vdash$	Pasta is cheap

Entailment : Hypothesis can be inferred from a given premise

Contradiction : Hypothesis cannot be inferred from a given premise

Neutral : undetermined

Natural Language Inference (NLI) Systems

In recent years...

The application of Large Language Models (LLMs) has become the mainstream approach for NLI [Lan et al., 2020; Raffel et al., 2020; He et al., 2021]

Natural Language Inference on RTE accuracy for natural language models
by https://paperswithcode.com/sota/natural-language-inference-on-rte

Natural Language Inference (NLI) Systems

In recent years…

The application of **Large Language Models (LLMs)** has become the mainstream approach for NLI [Lan et al., 2020; Raffel et al., 2020; He et al., 2021]

However…

Obtaining the inference process from LLMs is not straightforward

➡ Ensuring explainability is an active research area [Gunning et al., 2017]

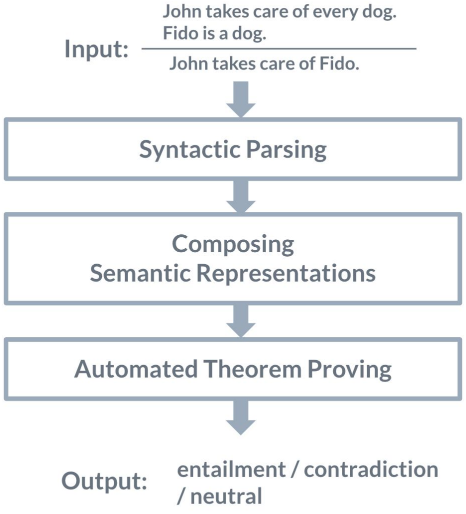
Natural Language Inference on RTE accuracy for natural language models
by https://paperswithcode.com/sota/natural-language-inference-on-rte

Symbolic Approaches to NLI

[Bos et al., 2004; Chatzikyriakidis and Luo, 2014; Martínez-Gómez et al., 2017; Chatzikyriakidis and Bernardy, 2019]

- Take the form of a linguistically-oriented pipeline consisting of syntactic parsers, semantic representations and theorem provers
- Can provide the inference process (explanation) as a logical formula
- Can represent complex linguistic phenomena regardless of how deep they get embedded in syntactic structures

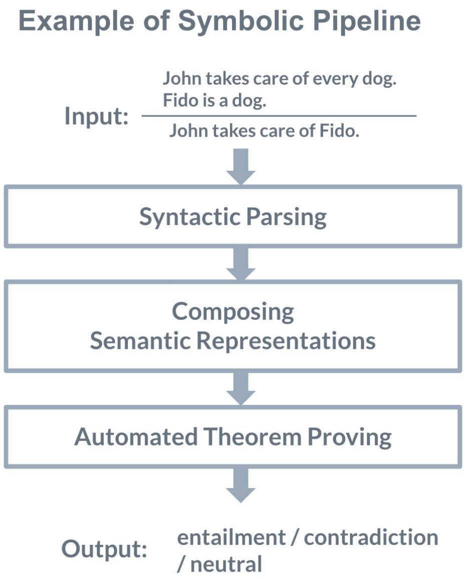
Example of Symbolic Pipeline



Symbolic Approaches to NLI

[Bos et al., 2004; Chatzikyriakidis and Luo, 2014; Martínez-Gómez et al., 2017; Chatzikyriakidis and Bernardy, 2019]

- Take the form of a linguistically-oriented pipeline consisting of syntactic parsers, semantic representations and theorem provers
- Can provide the inference process (explanation) as a logical formula
- Can represent complex linguistic phenomena regardless of how deep they become embedded in syntactic structures
- Cannot learn from large corpora or provide comparable representations



Fusion of neural and symbolic approaches

[Cooper, 2019] [Larsson, 2020]

Neural approaches

End-to-end systems with neural network

- ✓ Can learn from large corpora
- ✓ Can provide comparable representations
- ✗ Cannot provide explanations in a straightforward manner

Symbolic approaches

Pipeline systems with type-logical semantics

- ✓ Can be explained by using a logical formula
- ✓ Can address complex linguistic phenomena
- ✗ Do not possess learning capabilities and cannot provide comparable representations

➡ Fusion of neural and symbolic approaches can complement each other's weaknesses

Neural DTS [Bekki+, 2022; Bekki+, 2023]

- A fusion of neural and symbolic approaches
- An NLI system incorporating a deep neural network within dependent type semantics (DTS)
- A learning algorithm was proposed by Bekki et al. (2022) and the mathematical background was proposed by Bekki et al. (2023)
    - The implementation has been a remaining issue
    - It was not obvious ...
        - how the feed-forward neural classifiers would work with the type system

In this study, we...

- propose partial implementation for Neural DTS
  - neural classifiers
  - partial proof-search algorithm that integrates neural classifier
- evaluate and demonstrate the behavior of these implementations

2.

Neural DTS

[Bekki, Tanaka, Takahashi 2022; 2023]

The Fusion of Neural and Symbolic Approaches

Categories of the fusion approaches:

1. Emulating symbolic reasoning by embedding knowledge graphs, SAT problem and first-order logic
   [Guu+, 2015] [Das+, 2017] [Takahashi+, 2018] [Selsam+, 2019] [Demeester+, 2016] [Sourek+, 2018]
2. Introducing a similarity measure between symbols using distributional representations instead of symbols [Lewis+, 2013]
3. Using neural networks to control the direction of the proof search [Rocktäschel+, 2017] [Wang, 2017]
4. Embedding neural networks in symbolic reasoning [Cooper, 2019] [Larsson, 2020]
5. Choosing between symbolic and soft reasoning module depending on the characteristics of problems
   [Kalouli+, 2020]

The Fusion of Neural and Symbolic Approaches

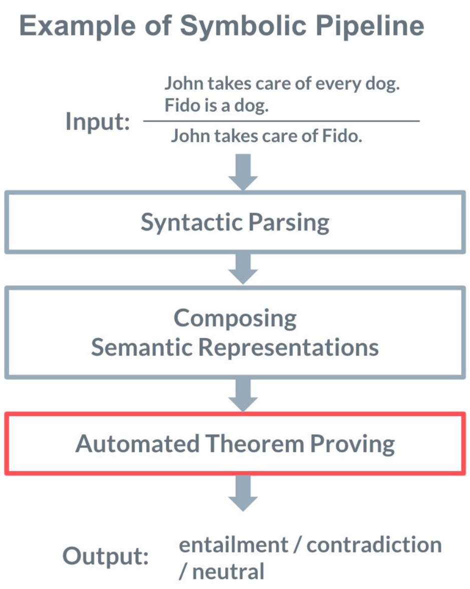
Approaches toward the fusion:

1. Emulating symbolic reasoning by embedding knowledge graphs, SAT problem and first-order logic
2. Introducing a similarity measure between symbols using distributional representations instead of symbols
3. Using neural networks to control the direction of the proof search
4. **Embedding neural networks in symbolic reasoning** [Cooper, 2019] [Larsson, 2020]
5. Choosing between symbolic and soft reasoning module depending on the characteristics of problems

Neural DTS

Neural DTS [Bekki, Tanaka, Takahashi 2022; 2023]

- Neural DTS embeds Neural classifiers in DTS
  - Logical framework : Dependent Type Semantics (DTS) [Bekki, 2014] [Bekki and Mineshima, 2017]
  - Names and predicates: Neural classifiers
- Neural DTS works with a syntactic parser and a semantic component that returns a semantic representations of the DTS
  - Input : semantic representations
  - Output : entailment / contradiction / neutral



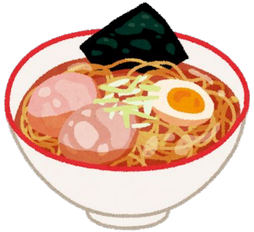
Neural DTS [Bekki, Tanaka, Takahashi 2022; 2023]

- Neural DTS embeds Neural classifiers in DTS
  - Logical framework : Dependent Type Semantics (DTS) [Bekki, 2014] [Bekki and Mineshima, 2017]
  - Names and Predicates: Neural classifiers
- Neur…
  frame…
  Mart…

- A unified theory of natural language semantics based on Martin-Löf type theory (MLTT) [Martin-Löf, 1984]
  - Anaphora and presuppositions as proof constructions
  - Implementations of type checker and prover [Bekki and Satoh, 2015] [Daido and Bekki 2020]
- Provide semantic representations as types in DTS

Example of Reasoning in Neural DTS

Ramen



Soba

Situation:

- Suppose you have heard of the names of two Japanese foods, "Ramen" and "Soba".
- You also know that the "Ramen" is a kind of noodle, and every noodle in Japan is cheap, but you are not sure whether "Soba" is also a noodle (although they have much in common).
- Then, your question is:
    - “Is Soba cheap?" given that "Every noodle is cheap."

Example of Reasoning in Neural DTS

Input : Every noodle is cheap $\vdash$ Soba is cheap

Example of Reasoning in Neural DTS

Input : Every noodle is cheap ⊢ Soba is cheap

Parsing and Semantic Composition

Semantic Representations : c:(x:e)->isNoodle(x)->isCheap(x) ⊢ c(soba)(s):isCheap(soba)

Semantic Representation

$$\dfrac{\dfrac{\text{c:(x:e)->isNoodle(x)->isCheap(x)} \qquad \dfrac{}{\text{soba:e}}\,(\text{CON})}{\text{c(soba):isNoodle(soba)->isCheap(soba)}}\,(\Pi E) \qquad \text{s:isNoodle(soba)}}{\text{c(soba)(s):isCheap(soba)}}\,(\Pi E)$$

Example of Reasoning in Neural DTS

Input : Every noodle is cheap ⊢ Soba is cheap

Parsing and Semantic Composition

Semantic Representations : c:(x:e)->isNoodle(x)->isCheap(x) ⊢ c(soba)(s):isCheap(soba)

this reasoning

$$\dfrac{\dfrac{\text{c:(x:e)->isNoodle(x)->isCheap(x)} \qquad \dfrac{}{\text{soba:e}}\,(\text{CON})}{\text{c(soba):isNoodle(soba)->isCheap(soba)}}\,(\Pi\text{E}) \qquad \text{s:isNoodle(soba)}}{\text{c(soba)(s):isCheap(soba)}}\,(\Pi\text{E})$$

It is not contained in the input

Example of Reasoning in Neural DTS

Knowledge Database of the System

isFood(ramen)	isFood(soba)
inBowl(ramen)	inBowl(soba)
isStapleFood(ramen)	isStapleFood(soba)
withSoup(ramen)	withSoup(soba)
isNoodle(ramen)	

☜ missing IsNoodle(soba)

Example of Reasoning in Neural DTS

Knowledge Database of the System

isFood(ramen)	isFood(soba)
inBowl(ramen)	inBowl(soba)
isStapleFood(ramen)	isStapleFood(soba)
withSoup(ramen)	withSoup(soba)
isNoodle(ramen)	

☜ missing isNoodle(soba)

We may expect that the system can guess that soba is also a noodle
This kind of abductive inference is one of motivations that neural DTS was developed

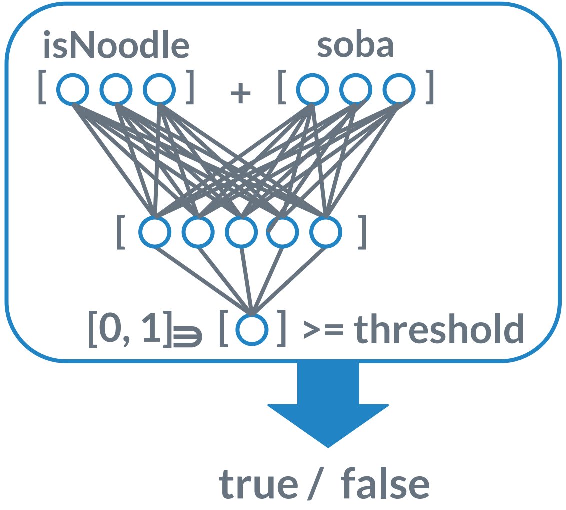
Example of Reasoning in Neural DTS

Knowledge Database of the System

isFood(ramen)	isFood(soba)
inBowl(ramen)	inBowl(soba)
isStapleFood(ramen)	isStapleFood(soba)
withSoup(ramen)	withSoup(soba)
isNoodle(ramen)	

train

Neural Classifier



Example of Reasoning in Neural DTS

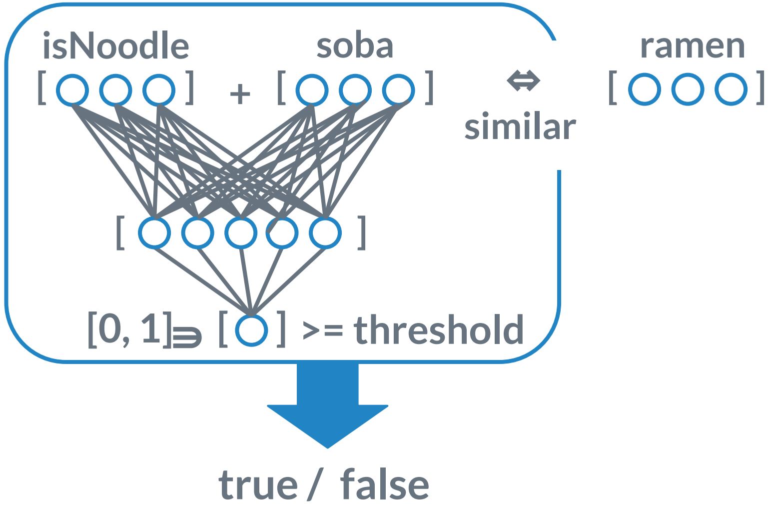
Knowledge Database of the System

isFood(ramen)
inBowl(ramen)
isStapleFood(ramen)
withSoup(ramen)
isNoodle(ramen)

isFood(soba)
inBowl(soba)
isStapleFood(soba)
withSoup(soba)

train

Neural Classifier



Example of Reasoning in Neural DTS

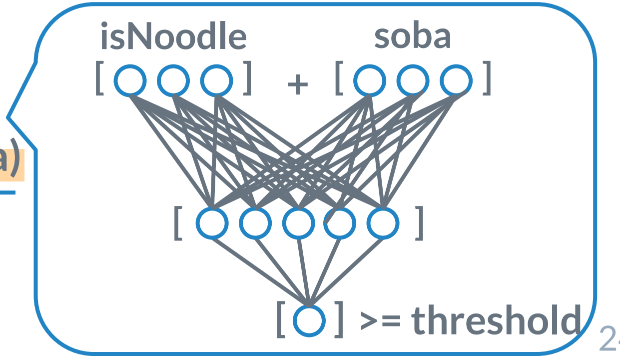
Input : Every noodle is cheap ⊢ Soba is cheap

Parsing and Semantic Composition

Semantic Representations : c:(x:e)->noodle(x)->cheap(x) ⊢ c(soba)(s):cheap(soba)

$$
\dfrac{\dfrac{c{:}(x{:}e)\text{->}isNoodle(x)\text{->}isCheap(x) \qquad \dfrac{}{soba{:}e}\,(CON)}{c(soba){:}isNoodle(soba)\text{->}isCheap(soba)}\,(\Pi E) \qquad s{:}isNoodle(soba)}{c(soba)(s){:}isCheap(soba)}\,(\Pi E)
$$

Call for Neural Classifier in Proof-Search



Example of Reasoning in Symbolic Approaches

Input :	Every noodle is cheap	⊢	Soba is cheap

Parsing and Semantic Composition

Semantic Representations :	c:(x:e)->noodle(x)->cheap(x)	⊢	c(soba)(s):cheap(soba)

(CON)

c:(x:e)->isNoodle(x)->isCheap(x) soba:e

(ΠE)

c(soba):isNoodle(soba)->isCheap(soba) (Axiom) s:isNoodle(soba)

(ΠE)

c(soba)(s):isCheap(soba)

Add Axiom

noodle(soba)
(soba is noodle)

Example of Reasoning in Symbolic Approaches

Input : Every noodle is cheap ⊢ Soba is cheap

Parsing and Semantic Composition

Semantic Representations : c:(x:e)->noodle(x)->cheap(x) ⊢ c(sob

Add Axiom

food(ramen) :	true
food(soba) :	true
inBowl(ramen) :	true
inBowl(soba) :	true
stapleFood(ramen) :	true
stapleFood(soba) :	true
withSoup(ramen) :	true
withSoup(soba) :	true
noodle(ramen) :	true

(CON)

c:(x:e)->isNoodle(x)->isCheap(x)   soba:e

(ΠE)

c(soba):isNoodle(soba)->isCheap(soba)   s:isNoodle(soba) ×

(ΠE)

c(soba)(s):isCheap(soba)

Example of Reasoning in Neural DTS

Every noodle is cheap $\vdash$ Soba is cheap

When the system need the knowledge that is not contained in the input (= Soba is noodle)

- Symbolic approaches
  - We need to explicitly add “Soba is noodle” as an axiom
- Neural DTS
  - Neural classifiers are called in proof-search
  - Neural classifiers are expected to learn Soba is similar to ramen which is a noodle even if corpus does not contain “Soba is noodle”
  - When “Soba is noodle” is fed to them, they return true

3.

Implementation of Neural DTS

Implementation of Neural DTS

- Neural DTS
  - Learning algorithm is proposed by Bekki et al. (2022) and the mathematical background is presented by Bekki et al. (2023)
  - The implementation of Neural DTS has been a remaining issue
- In this study
  - Implemented neural classifier
  - Implemented partial proof-search algorithm that integrates neural classifier

How is our implementation partial?

- The type theory of which we implemented proof-search algorithm is fragment of DTS
  - Our type theory only implemented non-dependent conjunction, disjunction and negation
  - However, in order to cover the semantic phenomena of natural language, dependent types play essential roles
- Our implementation is a partial implementation of neural DTS; The aim of this study is to show how we can fuse neural classifiers and type system

Integrating Neural Classifiers with Proof-search

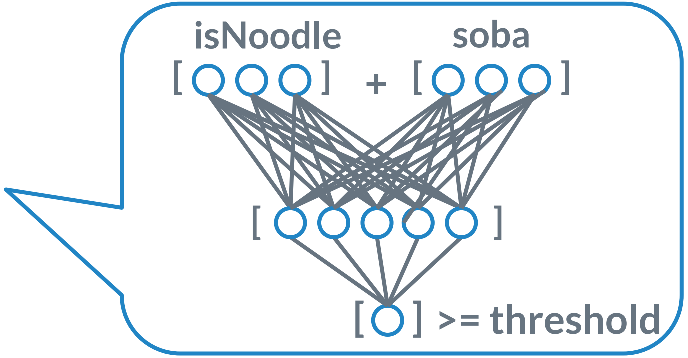
Atomic proposition

= "the name satisfies the predicate" ?

= Predicate(Name)

- Calls Neural Classifier when the proof-search algorithm encounters Predicate(Name)

w:isNoodle(soba)



Integrating Neural Classifiers with Proof-search

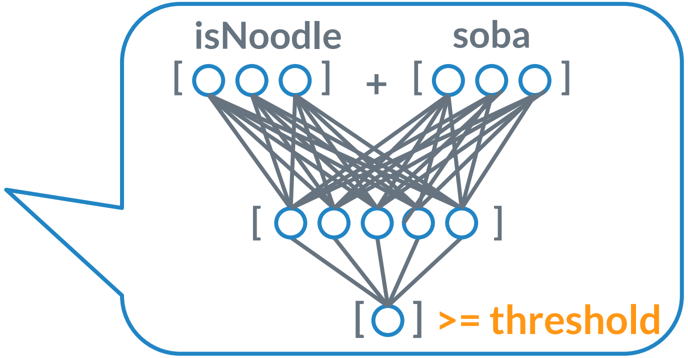
- Neural Classifier's Score of Predicate(Name) ≧ threshold

  → a term w has the type Predicate(Name)

  = Predicate(Name) has the proof w

  = the name satisfies the predicate

w:isNoodle(soba)

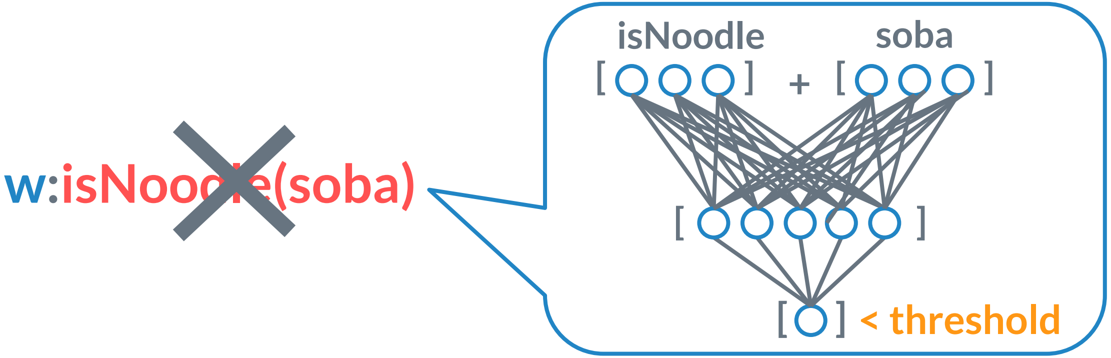


Integrating Neural Classifiers with Proof-search

- Neural Classifier's Score of Predicate(Name) < threshold

  → proposition Predicate(Name) does not have proof

  = the name doesn't satisfy the predicate

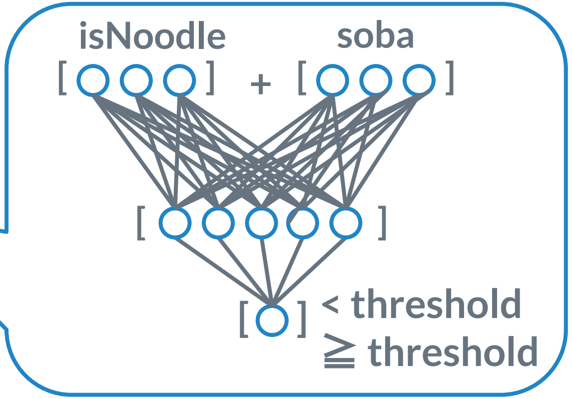


Integrating Neural Classifiers with Proof-search

Negation ¬A

- Negation of proposition A : expressed as A -> ⊥
- Score of Predicate(Name) < threshold
  → negative evidence = assumed to be of type
- Score of Predicate(Name) ≧ threshold
  → negative evidence = not to be of type

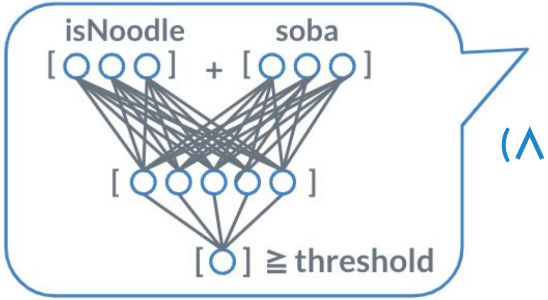
w:¬isNoodle(soba)



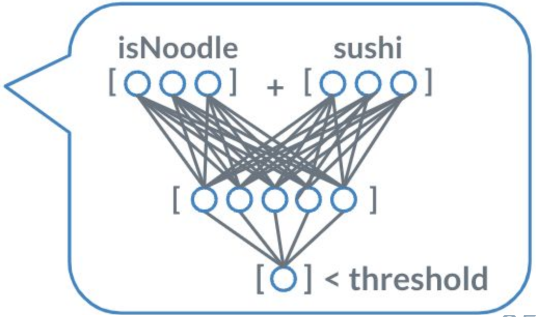
Integrating Neural Classifiers with Proof-search

Conjunction A ∧ B

- Treat as Pair type (non-dependent)
  - Pair type of A × B is the type whose terms are ordered pairs (m, n) with m : A and n : B
- If A has a proof and B has a proof → A ∧ B has a proof



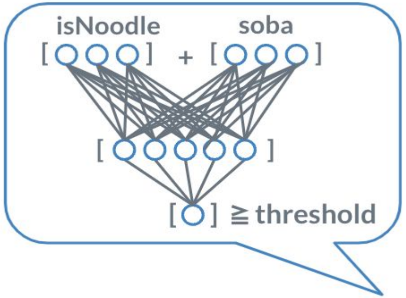
$$\dfrac{w+ : \mathrm{isNoodle(soba)} \qquad w- : \neg\mathrm{isNoodle(sushi}}{(w+, w-) : \mathrm{isNoodle(soba)} \wedge \neg\mathrm{isNoodle(sushi}}\,(\wedge I)$$

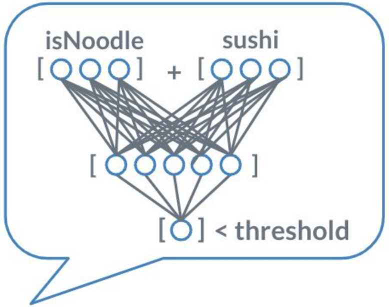


Integrating Neural Classifiers with Proof-search

Disjunction A ∨ B

- ▷ Treat as Sum type
  - ○ Sum type of A + B is the type whose terms are either terms m : A or terms n : B
- ▷ If A has a proof or B has a proof → A ∨ B has a proof





$$(\land \mathrm{I})\frac{w+ : \mathrm{isNoodle(soba)}}{(w+, w-) : \mathrm{isNoodle(soba)} \lor \neg \mathrm{isNoodle(sushi)}}$$

$$(\land \mathrm{I})\frac{w- : \neg \mathrm{isNoodle(sushi)}}{(w+, w-) : \mathrm{isNoodle(soba)} \lor \neg \mathrm{isNoodle(sushi)}}$$

4.
Experiments

Experiments

Issue : whether it was actually possible to fusion neural classifier and type system

→ Conducted two experiments to confirm this

① Check the accuracy

- The accuracy in atomic propositions is sufficient for future expansion in DTS

② Check the fusions of Neural Network and Logic

- Reason about complex propositions : Negation, Conjunction, Disjunction

Application of this proof-search algorithm

Apply this proof-search algorithm

→ Predicts whether a simple relation consisting of one predicate and two entities is valid

ex) a dog has a tail

- Follow [Richard Socher et al. 2013]
- Can we predict a relation that does not appear in the training data?

  ex) dog has a tail => puppy has a tail

Hyper Parameters

- Loss function : binaryCrossEntropyLoss
  Activation function : Sigmoid
- Hidden layer : 2 layers
- Threshold = 0.5
- Learning rate = 5e-2
- Use hasktorch

Experimental Data

Yago4 https://yago-knowledge.org/downloads/yago-4

- Large knowledge base
- About people, cities, countries, films, and organizations
- Triplet : (entity1, relation, entity2)

  Pair : (entity, class)
- 13,000,788 entities, 110 relations, 249 classes

Experimental Data

Negative data is generated from the same classes and relations as positive data

- Randomly generated triplets that are not included in the positive data
- Positive data and negative data have the same size

Experimental Data

- Use triplet (class1, relation, class2)
  - With extracting relationships between classes from Yago4
- Neural DTS need data on entity and must contain enough relationships with the predicate for each individual entity
- However such data could not be constructed in Yago4
  → so we created triplets using classes instead of entities
- It sufficient as a preliminary experiment

Example Experimental Data

(Windows_10, copyrightHolder, Microsoft)

- Classes of Windows_10 : [CreativeWork, Product, Thing]

  Classes of Microsoft : [Organization, Thing]

  →[CreativeWork, copyrightHolder, Organization],

  [Product, copyrightHolder, Organization] etc.

Experiment ①

- Binary classification to correctly predict relationships on triplets
- Multi-Layer Perceptron (MLP) & Neural Tensor Network(NTN)
- Epochs = 100
- Scores (on atomic propositions)

	accuracy	precision	recall	f1
MLP	0.9614	0.9517	0.9722	0.9618
NTN	0.9801	0.9710	0.9898	0.9803

- The accuracy is sufficient for future expansion in DTS

Examples of Successful Inference

- [Photograph,copyrightHolder,Person] : valid -> True
- [Chapter,copyrightHolder,Pond] : invalid -> False

- Not included in training dataset
  - Knowledge acquired by leaning
- Unknown atomic propositions

Examples of Failure Inference

- [MusicPlaylist, children, Person] : invalid -> True
- [Book, hasPart, Chapter] : valid -> False

- These relations have broader meaning than "copyrightHolder"

  → Predictions are likely to fail

Experiment ②

- Check whether this algorithm is able to infer correctly on its complex propositions
  - Negation, Conjunction, and Disjunction
- Negation
  - When the atomic proposition was true, all negative propositions were false (Vice versa)
- Conjunction, Disjunction
  - Use the inference results of each atomic proposition
    $\rightarrow$ Always correct when the atomic propositions are correct

Discussion ①

- End-to-end NN : Not relevant between atomic and other propositions
  - An atomic proposition outputs False vs. the negation may be also False
  - Two atomic propositions are True vs. the conjunction may be False

→ These relationships are logically incorrect

- Neural Type Checker : Relevant
  - An atomic proposition outputs False -> the negation is always True
  - Two atomic propositions are True -> the conjunction is always True

Discussion ②

- Symbolic reasoning : Treat external knowledge as a non logical axiom

  Neural DTS : Can learn and infer things not included in the original knowledge
- Improving the accuracy of the prediction of atomic propositions

  → Even more complex propositions can be predicted correctly

5.
Conclusion

Conclusion

- Neural DTS
  - Embedding DNN in DTS $\longrightarrow$ explainability & calculate similarity
- In this study
  - Partial implementation of Neural DTS
  - Neural classifier is called at atomic proposition
  - Experiments
    - Relation consisting of a predicate and two names
    - Negation, conjunction, and disjunction

Future Work

- Support more complex reasoning
  - Now : only negation, conjunction, and disjunction
  - Next : implement universal & existential quantification
    - Eventually accommodate complex sentence structures
      → take more advantage of symbolic inference
    - DTS is already implemented; use that implementation
- Implement Neural DTS
  - With an automatic theorem prover

Thank you

This research is supported by JST CREST project

"AI systems that can explain in language based on knowledge and reasoning"

References ①

- Bill MacCartney and Christopher D. Manning (2008) : Modeling Semantic Containment and Exclusion in Natural Language Inference
- Jun Lan, Jiwan Ge, Jinfang Yu, Sisi Shan, Huan Zhou, Shilong Fan, Qi Zhang, Xuanling Shi, Qisheng Wang, Linqi Zhang & Xinquan Wang (2020) : Structure of the SARS-CoV-2 spike receptor-binding domain bound to the ACE2 receptor
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu (2020) : Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer
- Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, Ross Girshick (2021) : Masked Autoencoders Are Scalable Vision Learners
- Martine P. Bos, Boris Tefsen, Jeroen Geurtsen, Jan Tommassen (2004) : Identification of an outer membrane protein required for the transport of lipopolysaccharide to the bacterial cell surface
- Stergios Chatzikyriakidis, Zhaohui Luo (2014) : Natural Language Inference in Coq
- Concepción Martínez-Gómez, Víctor M León, Susana Calles, Marina Gomáriz-Olcina, A Dick Vethaak (2017) : The adverse effects of virgin microplastics on the fertilization and larval development of sea urchins
- Stergios Chatzikyriakidis, Jean-Philippe Bernardy (2019) : A Wide-Coverage Symbolic Natural Language Inference System
- Lluís Màrquez, Chris Callison-Burch, Jian Su (2015) : Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing
- Rajarshi Das, Arvind Neelakantan, David Belanger, Andrew McCallum (2017) : Chains of Reasoning over Entities, Relations, and Text using Recurrent Neural Networks
- Ryo Takahashi, Ran Tian, Kentaro Inui (2018) : Interpretable and Compositional Relation Learning by Joint Training with an Autoencoder

References ②

- Daniel Selsam, Matthew Lamm, Benedikt Bünz, Percy Liang, Leonardo de Moura, David L. Dill (2019) : Learning a SAT solver from single-bit supervision
- Thomas Demeester, Tim Rocktäschel, Sebastian Riedel (2016) : Lifted Rule Injection for Relation Embeddings
- Gustav Sourek, Vojtech Aschenbrenner, Filip Zelezny, Steven Schockaert, Ondrej Kuzelka (2018) : Lifted relational neural networks: efficient learning of latent relational structures
- Combined Distributional and Logical Semantics (2013) : Mike Lewis, Mark Steedman
- Tim Rocktäschel, Sebastian Riedel (2017) : End-to-end differentiable proving
- Mingzhe Wang, Yihe Tang, Jian Wang, Jia Deng (2017) : Premise selection for theorem proving by deepgraph embedding
- Robin Cooper (2019) : Representing Types as Neural Events
- Staffan Larsson (2020) : Discrete and Probabilistic Classifier-based Semantics
- Aikaterini-Lida Kalouli, Richard Crouch, Valeria de Paiva (2020) : Hy-NLI: a Hybrid system for Natural Language Inference
- Daisuke Bekki (2014) : Representing Anaphora with Dependent Types
- Daisuke Bekki, Ribeka Tanaka, Yuta Takahashi (2022) : Integrating Deep Neural Networks with Dependent Type Semantics
- Daisuke Bekki, Ribeka Tanaka, Yuta Takahashi (2022) : Learning Knowledge with Neural DTS
- Robin Cooper (2005) : Records and record types in semantic theory
- Martin-Löf (1984) : Intuitionistic Type Theory

References ③

- Richard Socher, Danqi Chen, Christopher D. Manning, Andrew Y. Ng (2013) : Reasoning With Neural Tensor Networks for Knowledge Base Completion
- Andres Löh, Conor McBride, Wouter Swierstra (2010) : A Tutorial Implementation of a Dependently Typed Lambda Calculus
- Daisuke Bekki, Miho Satoh (2015) : Calculating Projections via Type Checking
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut (2020) : ALBERT: A Lite BERT for self-supervised learning of language representations
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2020) : Exploring the limits of transfer learning with a unified text-to-text transformer
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen (2021) : DEBERTA: Decoding Enhanced BERT with disentangled attention
- Ahmet, Ahmed, and Tariq Abdullah. "Real-time social media analytics with deep transformer language models: a big data approach." 2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE). IEEE, 2020.
- Gunning, David. "Explainable artificial intelligence (xai)." Defense advanced research projects agency (DARPA), nd Web 2.2 (2017): 1.