Paper review: RuleNN

Eszter Iklódi

TU Wien

Mar 24, 2022

Sen et al. (2020)

Published in EMNLP2020 (Empirical Methods in Natural Language Processing)

Learning Explainable Linguistic Expressions with Neural Inductive Logic Programming for Sentence Classification

Prithviraj Sen
IBM Research
San Jose, CA, USA
senp@us.ibm.com

Marina Danilevsky
IBM Research
San Jose, CA, USA
mdanile@us.ibm.com

Yunyao Li
IBM Research
San Jose, CA, USA
yunyaoli@us.ibm.com

Siddhartha Brahma
Google Research
Mountain View, CA, USA
sidbrahma@google.com

Matthias Boehm
Graz University of Technology
Graz, Austria
m.boehm@tugraz.at

Laura Chiticariu
IBM Watson
San Jose, CA, USA
chiti@us.ibm.com

Rajasekar Krishnamurthy
IBM Watson
San Jose, CA, USA
rajase@us.ibm.com

Brief summary

Topic
transparent learning

Task
sentence classification

Model
first-order logic rules

Method
neural network

Results
outperforms statistical and neuro-symbolic methods, comparable to RNN

Implication
explainable and editable models

Context

Background

- Black-box models: difficult-to-interpret, undesirable bias
- There is interest in interpretability.
- Attempts to explain
  - surrogate models
  - neural network layer activation (attention)

Knowledge gap

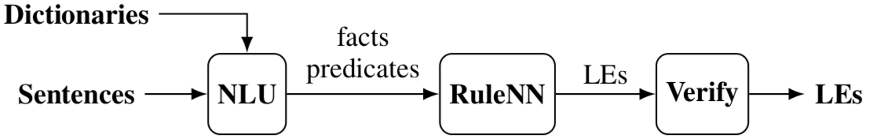
- Do surrogate models correctly reflect the process?
- Treat explainability as first-class citizen and not as an after-thought

Research question

- Is it possible to devise a neural network that directly learns a model expressed in a clear, human-readable dialect?

Suggested approach

Architecture

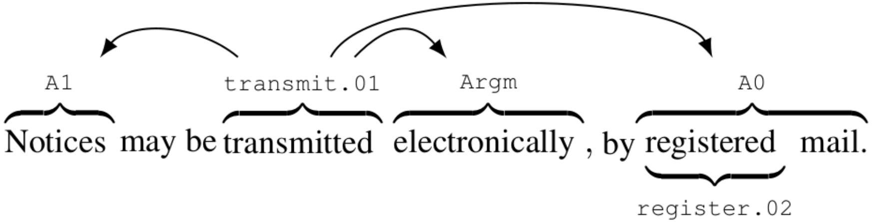


Idea

- Model NLP with first-order logic (FOL)
- Use shallow semantic parsing
- Combine the two into **linguistic expression (LE)**

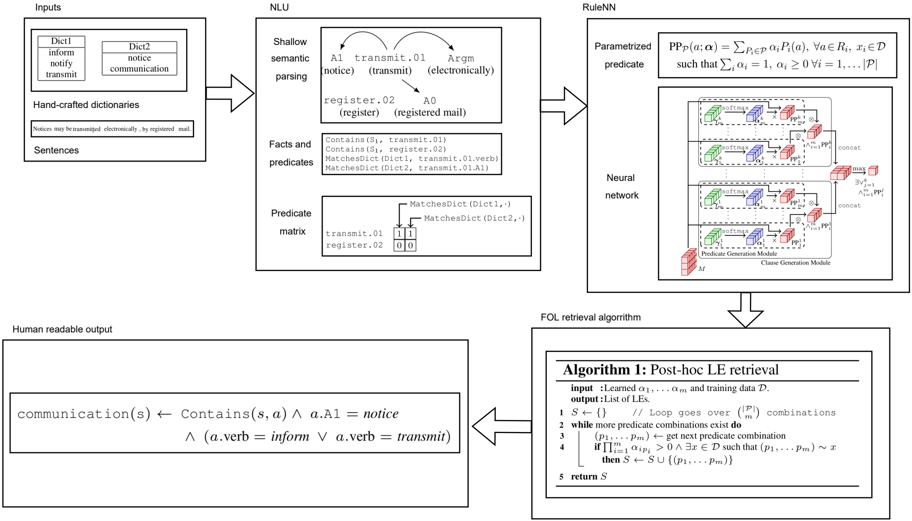
Linguistic Expression (LE) - Example

Shallow semantic parsing



Learned rule = Linguistic expression

$$\texttt{communication(s)} \leftarrow \texttt{Contains}(s, a) \wedge a.\texttt{A1} = \textit{notice}$$
$$\wedge \ (a.\text{verb} = \textit{inform} \ \vee \ a.\text{verb} = \textit{transmit})$$


Inputs
Dict1
inform
notify
transmit
Dict2
notice
communication
Hand-crafted dictionaries
Notices may be transmitted electronically , by registered mail.
Sentences
NLU
Shallow semantic parsing
A1 transmit.01 Argm
(notice) (transmit) (electronically)
register.02 A0
(register) (registered mail)
Facts and predicates
Contains(S1, transmit.01)
Contains(S1, register.02)
MatchesDict(Dict1, transmit.01.verb)
MatchesDict(Dict2, transmit.01.A1)
Predicate matrix
MatchesDict(Dict1,·)
MatchesDict(Dict2,·)
transmit.01 1 1
register.02 0 0
RuleNN
Parametrized predicate
$\mathrm{PP}_{\mathcal{P}}(a;\boldsymbol{\alpha}) = \sum_{P_i \in \mathcal{P}} \alpha_i P_i(a),\ \forall a \in R_i,\ x_i \in \mathcal{D}$
such that $\sum_i \alpha_i = 1,\ \alpha_i \geq 0\ \forall i = 1, \ldots |\mathcal{P}|$
Neural network
softmax
concat
max
Predicate Generation Module
Clause Generation Module
M
FOL retrieval algorrithm
Algorithm 1: Post-hoc LE retrieval
input : Learned $\alpha_1, \ldots \alpha_m$ and training data $\mathcal{D}$.
output : List of LEs.
1 $S \leftarrow \{\}$ // Loop goes over $\binom{|\mathcal{P}|}{m}$ combinations
2 while more predicate combinations exist do
3 $(p_1, \ldots p_m) \leftarrow$ get next predicate combination
4 if $\prod_{i=1}^{m} \alpha_{ip_i} > 0 \wedge \exists x \in \mathcal{D}$ such that $(p_1, \ldots p_m) \sim x$ then $S \leftarrow S \cup \{(p_1, \ldots p_m)\}$
5 return S
Human readable output
$\texttt{communication}(s) \leftarrow \texttt{Contains}(s, a) \wedge a.\texttt{A1} = \textit{notice}$
$\wedge\ (a.\text{verb} = \textit{inform} \vee a.\text{verb} = \textit{transmit})$

Shallow semantic parsing

Syntactic structures - Penn Treebank (Marcinkiewicz, 1994)

```
(S (NP I)
   (VP have
       (NP a dog)
       (PP from
           (NP a Hungarian shelter))))
```

Semantic structures - PropBank (Palmer et al., 2005)

- Proposition Bank
- Penn Treebank structures + semantic role labels
- Shallow: no higher-order phenomena, e.g. coreference
- Predicate-argument information
    - ArgNr: numbered arguments; can be mapped to any theory
    - ArgM: adjunct-like arguments, like location, cause, time etc.

PropBank notation

Frameset **decline.01** "go down incrementally"

Arg1: entity going down

Arg2: amount gone down by, EXT

Arg3: start point

Arg4: end point

Ex: . . . [$_{\text{Arg1}}$ its net income] *declining* [$_{\text{Arg2-EXT}}$ 42%] [$_{\text{Arg4}}$ to $121 million] [$_{\text{ArgM-TMP}}$ in the first 9 months of 1989]. (wsj_0067)

Frameset **decline.02** "demure, reject"

Arg0: agent

Arg1: rejected thing

Ex: [$_{\text{Arg0}}$ A spokesman$_i$] *declined* [$_{\text{Arg1}}$ *trace*$_i$ to elaborate] (wsj_0038)

First-order logic (FOL)

First order logic // Predicate logic // Quantificational logic

Definition

- $\exists$ existential quantification
- $\forall$ universal quantification
- $\wedge$ conjunction
- $\vee$ disjunction
- $\neg$ negation

How to learn?

- already done: inductive logic programming (ILP), statistical learning (StarAI), neuro-symbolic AI
- knowledge gap: none of these target NLP so far

Linguistic Expression (LE) - Explanation

Head predicate (label)

Rule a.k.a. LE $\boxed{R_3}$: $\boxed{\texttt{communication}(s)} \leftarrow \boxed{\texttt{Contains}(s, a)}$ Distinguised predicate for LE

$\wedge\ (\boxed{a.\texttt{A0 contains } \textit{notice}}$ Semantic predicate

$\vee\ \boxed{a.\texttt{A0 contains } \textit{communication}})$

$\wedge\ \boxed{a.\text{tense} = \texttt{future}}$ Syntactic predicate

S_3 : $\overbrace{\textit{Notices}\text{ required in writing under this agreement}}^{\texttt{A0}}$ will $\overbrace{\text{be made}}^{\texttt{be.01}}$ to the appropriate contact(s) ..

Linguistic Expression (LE)

Logical Predicate

- Boolean-valued function returning true or false.
- Atom: a predicate whose variables denote constants.
- Fact: an atom that holds true.

SRL (semantic role labeling) Predicate

- True, if the given dictionary **contains** the action's surface form corresponding to a specific *semantic attribute*.
- Example semantic attributes: *verb*, *A0*, *A1*, *ARGM*

Syntactic Predicate

- True if action's value **is equal to** a specific value of a *syntactic attribute*.
- Example syntactic attributes: *tense*, *voice*

Linguistic Expression (LE)

Rule

- Head predicate: label
- Body predicate: conditions

LE

- A clause over a sentence and action
- distinguished *Contains* predicate + SRL and syntactic predicates

Linguistic Expression (LE) - Explanation

Head predicate (label)

Rule a.k.a. LE $\boxed{R_3}$: $\boxed{\texttt{communication}(s)} \leftarrow \boxed{\texttt{Contains}(s, a)}$ Distinguised predicate for LE

$\wedge \; (\boxed{a.\texttt{A0 contains } \textit{notice}}$ Semantic predicate

$\quad \vee \boxed{a.\texttt{A0 contains } \textit{communication}})$

$\wedge \boxed{a.\text{tense} = \texttt{future}}$ Syntactic predicate

S_3 : $\overbrace{\textit{Notices}\text{ required in writing under this agreement}}^{\texttt{A0}}$ will $\overbrace{\text{be made}}^{\texttt{be.01}}$ to the appropriate contact(s) ..

Task 1: TREC - question classification

Labels

ABBREVIATION	abbreviation
abb	abbreviation
exp	expression abbreviated

Examples

ABBR:exp What is the full form of .com ?
HUM:title What is the oldest profession ?
DESC:def What are liver enzymes ?

Label distribution

Label	Skew	$\|\mathcal{P}\|$	
LOC	0.18	133	Location
HUM	0.28	109	Human
NUM	0.17	127	Numeric
ENTY	0.22	137	Entity
DESC	0.22	122	Description
ABBR	0.02	38	Abbreviation

Task 2: Contracts

Task

- IBM proprietary data
- Multi-label task, but treated as binary classification
- Sentences from legal contracts among enterprise
- Training set: IBM first-party
- Test set: diverse companies
- Generalization?

Label	Skew	$\lvert\mathcal{P}\rvert$
W	0.09	101
SoW	0.07	48
DR	0.06	80
IP	0.05	79
C	0.06	39
P&T	0.10	117
T&T	0.08	77
P&B	0.05	95
L	0.04	71

Labels

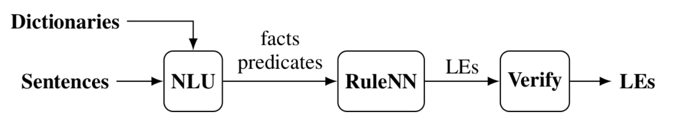
Payment Terms & Billing	Elements that detail how and when a party is to pay or get paid, as well as the items or fees the parties are paying or billed for. Includes references to modes of payment or payment mechanisms.
Pricing & Taxes	Elements that refer to specific amounts or figures that are associated with individual deliverables that are exchanged (for example, how much something costs) as part of satisfying the terms of the contract. Includes references to specific figures or methods for calculating prices or tax amounts.

RuleNN architecture

Steps to take

1. Learn discriminative predicates, a.k.a. FOL rules
2. Combine them into LEs
3. Learn multiple LEs

Architecture



RuleNN

1. PGM: Predicate Generation Module
2. CGM: Clause Generation Module

NLU

- input: sentences and dictionaries
- dictionaries: hand-crafted for each label
- output: generated facts and predicates

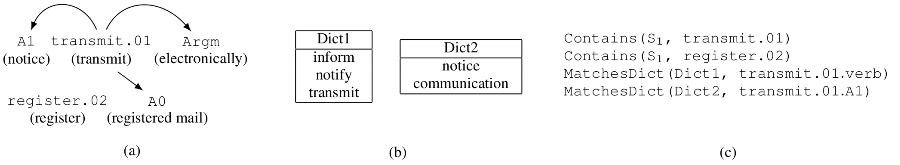


Figure 3: Generating (c) facts and predicates from (a) shallow semantic parsing and (b) dictionaries.

Task formulation: Input representation

Predicate matrix

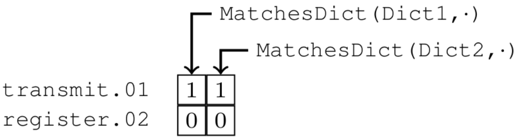
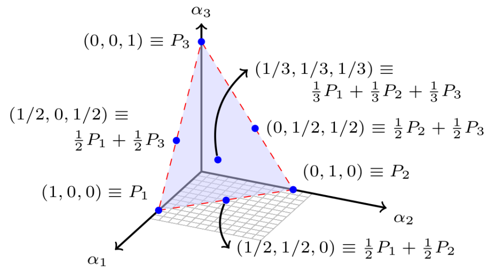


Figure 4: Example predicate matrix where rows and columns denote actions and predicates, respectively.

$$M \in \{0, 1\}^{|\mathcal{R}_i| \times |\mathcal{P}|}$$

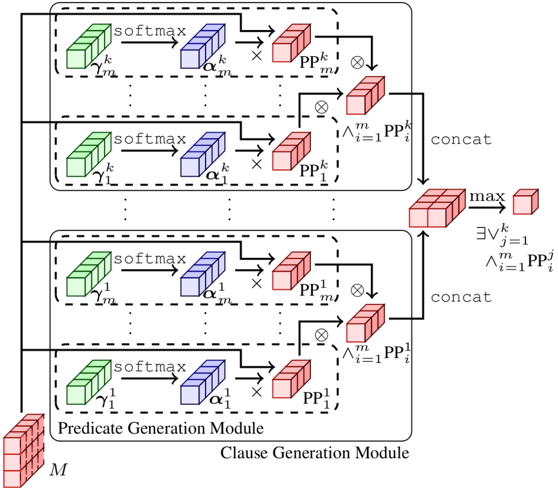
Task formulation: Learnable parameters

Parametrized predicate (PP)



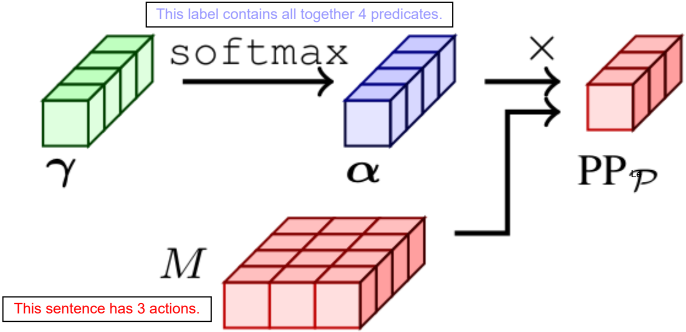
- linear (convex) combination of predicates
- one hot encoding $\rightarrow$corner of the convex hull
- update: backpropagation

$$\mathrm{PP}_{\mathcal{P}}(a;\boldsymbol{\alpha}) = \textstyle\sum_{P_i \in \mathcal{P}} \alpha_i P_i(a),\ \forall a \in R_i,\ x_i \in \mathcal{D}$$
$$\text{such that} \textstyle\sum_i \alpha_i = 1,\ \alpha_i \geq 0\ \forall i = 1, \dots |\mathcal{P}|$$



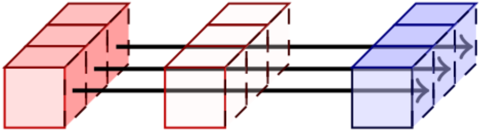
softmax
γ_m^k
α_m^k
$\times$
PP_m^k
$\otimes$
softmax
γ_1^k
α_1^k
$\times$
PP_1^k
$\otimes$
$\wedge_{i=1}^m PP_i^k$
concat
max
$\exists \vee_{j=1}^k \wedge_{i=1}^m PP_i^j$
softmax
γ_m^1
α_m^1
$\times$
PP_m^1
$\otimes$
concat
softmax
γ_1^1
α_1^1
$\times$
PP_1^1
$\otimes$
$\wedge_{i=1}^m PP_i^1$
Predicate Generation Module
Clause Generation Module
M

Method: Predicate Generation Module (PGM)



Name	Global/Local	Description
M	Local	Predicate matrix for instance
$\mathcal{R}$	Local	Actions in instance
PP_i^j	Local	Responses for actions belonging to instance
α_i^j	Global	Attention weights defining a learned predicate
γ_i^j	Global	Log attention weights for learned predicate
k	Global	Number of LEs (hyperparameter)
m	Global	Length of LEs (hyperparameter)

Method: Clause Generation Module (CGM)



$\mathrm{PP}^1_{\mathcal{P}} \otimes \mathrm{PP}^2_{\mathcal{P}} \qquad \mathrm{PP}^1_{\mathcal{P}} \wedge \mathrm{PP}^2_{\mathcal{P}}$

Name	Global/Local	Description
M	Local	Predicate matrix for instance
$\mathcal{R}$	Local	Actions in instance
PP_i^j	Local	Responses for actions belonging to instance
α_i^j	Global	Attention weights defining a learned predicate
γ_i^j	Global	Log attention weights for learned predicate
k	Global	Number of LEs (hyperparameter)
m	Global	Length of LEs (hyperparameter)

- Conjunction is replaced by differentiable, element-wise product t-norm.
- Combining different LEs: max across all CGM outputs. It can then be compared to the label of the sentence.

Method: Big picture

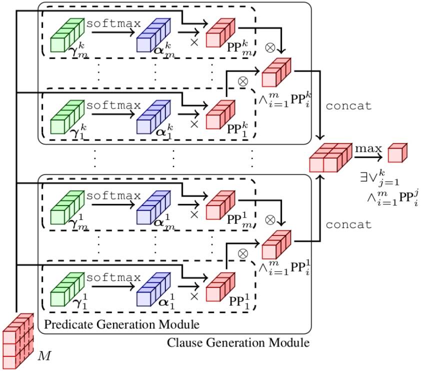


Figure 7: RuleNN for learning k m-length clauses.

Name	Global/Local	Description
M	Local	Predicate matrix for instance
$\mathcal{R}$	Local	Actions in instance
PP_i^j	Local	Responses for actions belonging to instance
α_i^j	Global	Attention weights defining a learned predicate
γ_i^j	Global	Log attention weights for learned predicate
k	Global	Number of LEs (hyperparameter)
m	Global	Length of LEs (hyperparameter)

Method: Retrieve LEs expressed as FOL

1. Consider each m-combination of predicates.
2. Return LE if:
   - associated weight is non-zero
   - evaluates true on some sentences

Algorithm 1: Post-hoc LE retrieval

input : Learned $\alpha_1, \ldots \alpha_m$ and training data $\mathcal{D}$.
output : List of LEs.

1 $S \leftarrow \{\}$ `// Loop goes over` $\binom{|\mathcal{P}|}{m}$ `combinations`
2 **while** more predicate combinations exist **do**
3 $\quad (p_1, \ldots p_m) \leftarrow$ get next predicate combination
4 $\quad$ **if** $\prod_{i=1}^{m} \alpha_{ip_i} > 0 \land \exists x \in \mathcal{D}$ such that $(p_1, \ldots p_m) \sim x$ **then** $S \leftarrow S \cup \{(p_1, \ldots p_m)\}$
5 **return** S

Method: Algorithmic details

General

- RuleNN learns k LEs containing *up to* m PPs each.
- skewed data: negative sampling
- dropout before max-pooling
- exponential in m, but efficient for small m

Setup

- k=50, m=4
- dropout=0.5, batchsize=64, stepsize=0.01
- SGD with momentum=0.9

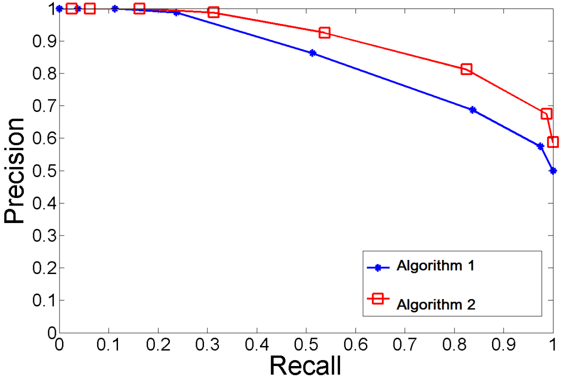
Comparison: Overview

Method	Category	Explainability	Input
MG	inductive logic programming	white-box	predicate-based
MG_{NT}	inductive logic programming	white-box	predicate-based
LSM	Markov logic network	white-box	predicate-based
BSRL	Markov logic network	white-box	predicate-based
MITI	multiple instance learning	white-box	predicate-based
MIRI	multiple instance learning	white-box	predicate-based
NeralLP	neuro-symbolic AI	white-box	predicate-based
RuleNN	neuro-symbolic AI	white-box	predicate-based
MINet	deep neural network	black-box	predicate-based
BiLSTM	deep neural network	black-box	token-based

Comparison: Metric

AUC-PR (area under the precision-recall curve)

- average of precision scores calculated for each recall threshold
- useful for imbalanced data



Background: Inductive logic programming (ILP)

Inductive logic programming - Muggleton (1991)

- Input: background knowledge (B) and examples (E)
- Output: logic program that entail ALL positive and NONE of the negative examples

Used implementations

Metagol (MG) - Cropper and Muggleton (2015)

- top-down approach (generates rules before testing them on data)
- generates only 0-error rules

Noise-tolerant metagol ($\mathrm{MG}_{\mathrm{NT}}$) - Muggleton et al. (2018)

- minimized error instead of 0-error rules,
- more suited for noisy real-world data

Background: Markov logic network (MLN)

Markov logic network - Richardson and Domingos (2006)

- Markov-network: like Bayesian-network, but undirected and may be cyclic
- combines first order logic and probabilistic graphical methods
- each formula/clause has an attached weight
- A highly expressive representation!

Used implementations

LSM (Learning using Structural Motifs) - Kok and Domingos (2010)

- first, search for clauses and then learn weights

BSRL (Boost Statistical Relational Learning) - Khot et al. (2011)

- learns weights and the structure of the MLN simultaneously

Background: Multiple instance learning (MIL)

Multiple instance learning - Dietterich et al. (1997)

- dataset: bag of instances
- classification: for each bag, whether it contains at least one positive instance or not
- sentences are bags of actions
- label is assigned if there exists at least one action, for which the learned predicates hold true. $\rightarrow$RuleNN is also MIL

Used implementations

MITI (multi-instance tree inducer) - Blockeel et al. (2005)

- learns an *instance* (not *bag*) tree classifier with best-first node expansion strategy

MIRI (multi-instance rule induction) - Bjerring and Frank (2011)

- output is more naturally a set of classification rules (if-then rules)

Background: Neuro-symbolic AI

Neuro-symbolic AI

- traditional rules-based AI approaches with modern deep learning techniques

Used implementations

NeuralIP - Yang et al. (2017)

- problem: learning of probabilistic first-order logical rules
- end-to-end parameter and structure learning

Background: Black-box methods

Used implementations

MINet - Wang et al. (2018)

- deep neural network with fully connected layers
- used for multiple instance learning

BiLSTM

- replaces tokens with GloVe embeddings
- bi-directional LSTM
- label: aggregation of hidden LSTM layers
- hidden layer units: 200–600
- an order of magnitude larger parameter set than RuleNN

Comparison: Experiments summary

	Label	MG$^{\square}$	MG$_{\text{NT}}^{\square}$	MITI$^{\square}$	MIRI$^{\square}$	MINet$^{\blacksquare}$	LSM$^{\square}$	BSRL$^{\square}$	NeuralLP$^{\square}$	*RuleNN*$^{\square}$	BiLSTM$^{\blacksquare}$
		Predicate-based								(↓ours↓)	
CONTRACTS	W	*NR*	0.07	0.184	0.156	0.294	–	0.183	0.537	0.685	**0.805 ± 0.010**
	SoW	*NR*	*NR*	0.011	0.011	0.018	–	0.015	0.438	0.658	**0.689 ± 0.030**
	DR	*NR*	0.05	0.144	0.147	0.258	–	0.021	0.614	**0.848**	0.807 ± 0.030
	IP	*NR*	0.13	0.145	0.153	0.244	–	0.148	0.550	**0.844**	0.787 ± 0.050
	C	*NR*	0.38	0.157	0.149	0.580	–	0.545	0.574	**0.788**	0.653 ± 0.020
	P&T	*NR*	0.30	0.111	0.083	0.314	–	0.269	0.516	0.813	**0.802 ± 0.030**
	T&T	*NR*	0.11	0.406	0.372	0.586	–	0.591	0.560	0.837	**0.846 ± 0.020**
	P&B	*NR*	0.10	0.111	0.122	0.154	–	0.192	0.533	**0.819**	0.786 ± 0.010
	L	*NR*	0.07	0.115	0.126	0.12	–	0.205	0.464	0.750	**0.741 ± 0.070**
TREC	LOC	*NR*	*NR*	0.710	0.699	0.833	0.473	0.835	0.470	0.904	**0.998 ± 0.001**
	HUM	*NR*	0.36	0.771	0.770	0.922	0.565	0.927	0.558	0.912	**0.999 ± 0.000**
	NUM	*NR*	*NR*	0.687	0.680	0.821	0.497	0.756	0.497	0.856	**0.996 ± 0.004**
	ENTY	*NR*	*NR*	0.365	0.373	0.591	0.481	0.425	0.576	0.745	**0.957 ± 0.020**
	DESC	*NR*	0.52	0.331	0.334	0.540	0.498	0.519	0.437	0.789	**0.995 ± 0.003**
	ABBR	*NR*	*NR*	0.731	0.731	0.688	0.542	0.735	0.443	0.774	**1.000 ± 0.000**

(c) *NR* and – denote no LEs learned and non-convergence, resp. Bold-font and underscore denotes best performing approach and best predicate-based method, resp. RuleNN (ours) is the best predicate-based method. Variation of BiLSTM's AUC-PR due to changing hidden dimensions is shown after ±.

Experiment: Explainability

No objective measure for LE's explainability :(

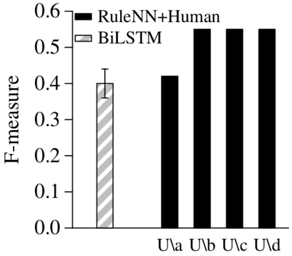
GUI

- filtering and ranking LEs based on precision and recall
- dropping/adding predicates
- instant re-evaluation

Experiment: Human-in-the-loop learning

Imitate iterative and collaboraive development

- 188 LEs learned for C (*Communication* label in *Contracts*)
- 4 data scientists with knowledge of FOL and NLU
- in half an hour time 6–8 LE selection
- →with human expertise LEs can be made smaller and more interpretable
- 4 explainable models of each subset of 3 participants



Experiment: Human-in-the-loop learning

Learned rule

$$\begin{aligned} R_3 : &\texttt{communication}(s) \leftarrow \texttt{Contains}(s, a) \\ &\wedge (a.\texttt{A0 contains } \textit{notice} \\ &\quad \vee a.\texttt{A0 contains } \textit{communication}) \\ &\wedge a.\text{tense} = \texttt{future} \end{aligned}$$

$\texttt{S}_3$: $\overbrace{\textit{Notices}\text{ required in writing under this agreement}}^{\texttt{A0}}$ will $\overbrace{\text{be made}}^{\texttt{be.01}}$ to the appropriate contact(s) ..

Experiment: Human-in-the-loop learning

Edited rule

$$
\begin{aligned}
R_3 :& \texttt{communication}(s) \leftarrow \texttt{Contains}(s, a) \\
& \wedge (a.\texttt{A0 contains } \textit{notice} \\
& \quad \vee a.\texttt{A0 contains } \textit{communication}) \\
& \cancel{\wedge\, a.\text{tense} = \texttt{future}}
\end{aligned}
$$

$\texttt{S}_3$: $\overbrace{\textit{Notices}\text{ required in writing under this agreement}}^{\texttt{A0}}$ will $\overbrace{\text{be made}}^{\texttt{be.01}}$ to the appropriate contact(s) ..

Discussion

Context

- A proposed neuro-symbolic learning method for sentence classification.
- Compared to similar methods: better efficiency and quality.

Key findings

- Yes, it is possible to learn human-interpretable models by designing NN-s with explainability in mind.

Strength and limitations

- It can be used for any MIL tasks assuming the predicates are given.
- It can learn rules combining any previously built classifier's output probabilities.

Disclaimer

- approach is different from explainable AI (!)

What's next?

- Learn rules on top of embeddings
- Learn rules and dictionaries jointly

Relevance to my research

- BRISE data is highly structured $\rightarrow$great potential for rule learning
- Current rules are defined as graph or text patterns.
- Potential experiment: change UD/4lang semantic representation to a shallow semantic predicate-argument representation?
- Challenges: PropBank for German?
  https://github.com/System-T/UniversalPropositions

...still unclear

- How dictionaries are constructed? (For TREC: "automatically capture surface forms that discriminate well among labels")
- How is the semantic/syntactic parser implemented?
- How are the facts and predicates constructed, given the dictionaries and shallow-semantic notations?
- How can we get an OR-relation within one SRL predicate? (I guess it's just a syntactic sugar for "contains" in case the dictionary has more than one element.)
- Why is it good to have multiple CGMs ($k > 1$)? The actual number of LEs is coming out as a result of the post-hoc algorithm.
- Generally, the term LE is a bit overloaded.

References

Luke Bjerring and Eibe Frank. Beyond trees: Adopting miti to learn rules and ensemble classifiers for multi-instance data. In *Australasian Joint Conference on Artificial Intelligence*, pages 41–50. Springer, 2011.

Hendrik Blockeel, David Page, and Ashwin Srinivasan. Multi-instance tree learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 57–64, 2005.

Andrew Cropper and Stephen H Muggleton. Logical minimisation of meta-rules within meta-interpretive learning. In *Inductive Logic Programming*, pages 62–75. Springer, 2015.

Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71, 1997.

Tushar Khot, Sriraam Natarajan, Kristian Kersting, and Jude Shavlik. Learning markov logic networks via functional gradient boosting. In *2011 IEEE 11th international conference on data mining*, pages 320–329. IEEE, 2011.

Stanley Kok and Pedro M Domingos. Learning markov logic networks using structural motifs. In *ICML*, 2010.

Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Using Large Corpora*, page 273, 1994.

Stephen Muggleton. Inductive logic programming. *New generation computing*, 8(4):295–318, 1991.

Stephen Muggleton, Wang-Zhou Dai, Claude Sammut, Alireza Tamaddoni-Nezhad, Jing Wen, and Zhi-Hua Zhou. Meta-interpretive learning from noisy images. *Machine Learning*, 107(7):1097–1118, 2018.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106, 2005.

Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1):107–136, 2006.

Prithviraj Sen, Marina Danilevsky, Yunyao Li, Siddhartha Brahma, Matthias Boehm, Laura Chiticariu, and Rajasekar Krishnamurthy. Learning explainable linguistic expressions with neural inductive logic programming for sentence classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4211–4221, 2020.

Xinggang Wang, Yongluan Yan, Peng Tang, Xiang Bai, and Wenyu Liu. Revisiting multiple instance neural networks. *Pattern Recognition*, 74:15–24, 2018.

Fan Yang, Zhilin Yang, and William W Cohen. Differentiable learning of logical rules for knowledge base reasoning. *Advances in neural information processing systems*, 30, 2017.

Thank you for you attention :)