A Logic-Driven Framework for Consistency of Neural Models

Tao Li, Vivek Gupta, Maitrey Mehta, Vivek Srikumar
School of Computing, University of Utah

EMNLP 2019
Arxiv:1909.00126
https://www.aclweb.org/anthology/D19-1405/

Presented by Jiayao Zhang
Feb 08, 2021

Knowledge as Supervision

- Example: Guiding (Semi-Supervised) Learning with Constraints

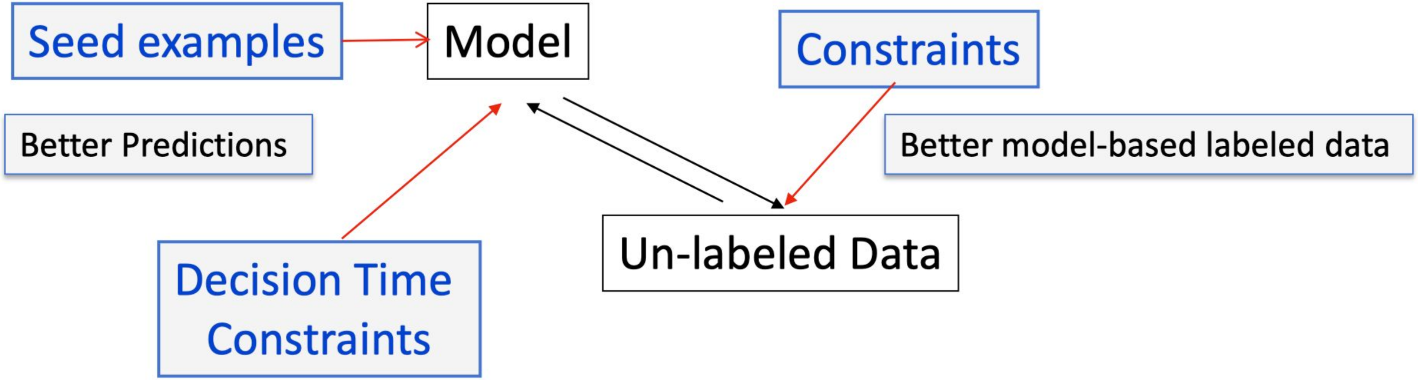


Figure Credit: Dan Roth. CIS-620 (Spring 2021) Lecture Notes 1.

Knowledge as Supervision

- Example: Guiding (Semi-Supervised) Learning with Constraints

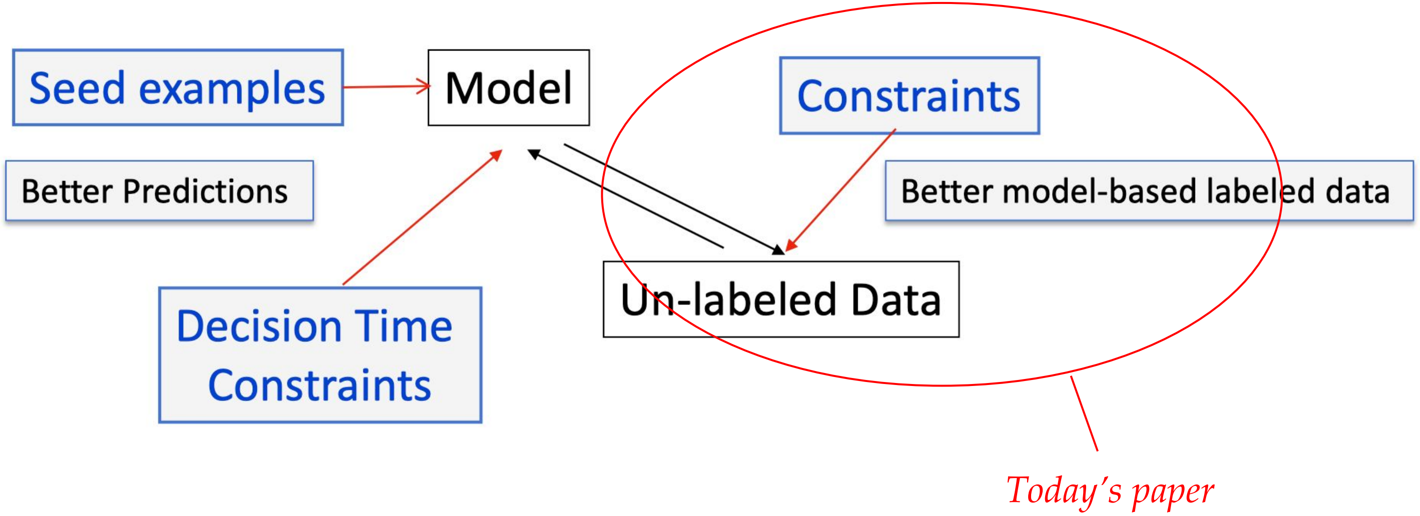


Figure Credit: Dan Roth. CIS-620 (Spring 2021) Lecture Notes 1.

Focus: Text Entailment Task

- Text Entailment Task
  - Give a two sentences, predict a label among *Entailment*, *Contradiction*, or *Neutral*.
  - Annotated Natural Language Inference (NLI) datasets from Amazon Turk such as SNLI, MultiNLI.

Sentence 1	**Sentence 2**	**Judgement**
A dog is *running* in the sand	The dog is *sitting* patiently	Contradiction
A woman *in black pants* is looking at her *cellphone*	a woman is looking at her phone	Entailment
A cyclist rides down a rocky mountain	He is an experienced rider	Neutral

The Stanford NLI Corpus (SNLI): https://nlp.stanford.edu/projects/snli/

Example

- Consider the sentences:
  - A: *Bob is **on a train** to Berlin*
  - B: *Bob is **traveling** to Berlin*
  - C: *Bob is **eating** in Berlin*

Example

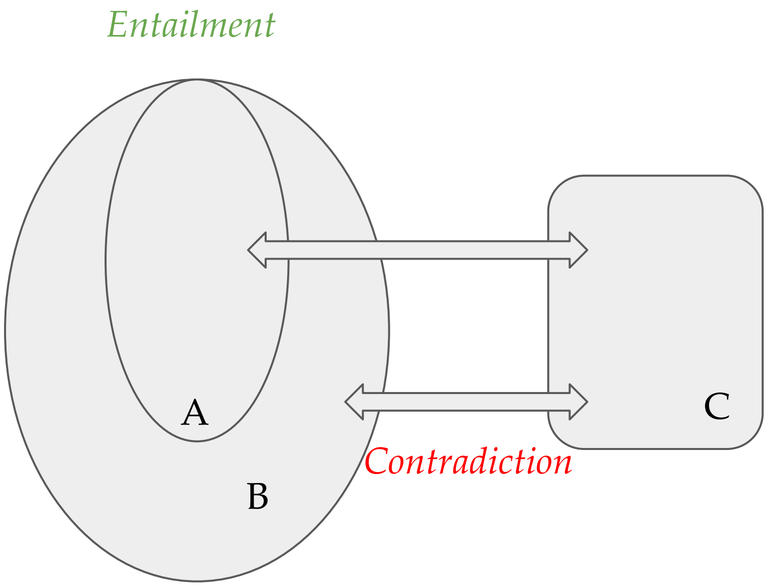
- Consider the sentences:
  - A: *Bob is* ***on a train*** *to Berlin*
  - B: *Bob is* ***traveling*** *to Berlin*
  - C: *Bob is* ***eating*** *in Berlin*
- We know -
  - (A, B): *Entailment*
  - (B, C): *Contradiction*

Example: Human Reasoning

- Consider the sentences:
  - A: *Bob is **on a train** to Berlin*
  - B: *Bob is **traveling** to Berlin*
  - C: *Bob is **eating** in Berlin*
- We know -
  - (A, B): *Entailment*
  - (B, C): *Contradiction*
  - Without even looking at (A, C) we can reason that (A, C) is ___________.

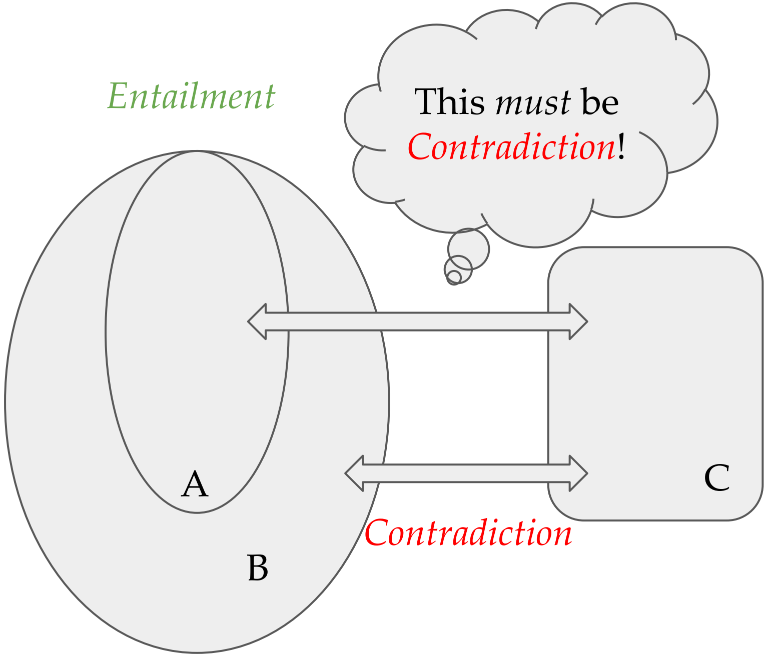
Example: Human Reasoning

- Consider the sentences:
  - A: *Bob is* ***on a train*** *to Berlin*
  - B: *Bob is* ***traveling*** *to Berlin*
  - C: *Bob is* ***eating*** *in Berlin*
- We know -
  - (A, B): *Entailment*
  - (B, C): *Contradiction*
  - Without even looking at (A, C) we can reason that (A, C) is ____________.



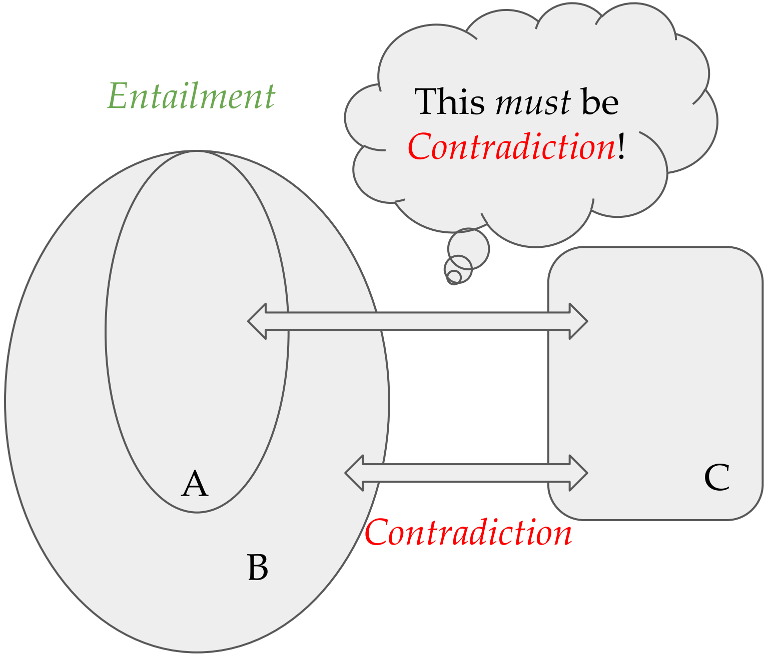
Example: Human Reasoning

- Consider the sentences:
  - A: *Bob is* ***on a train*** *to Berlin*
  - B: *Bob is* ***traveling*** *to Berlin*
  - C: *Bob is* ***eating*** *in Berlin*
- We know -
  - (A, B): *Entailment*
  - (B, C): *Contradiction*
  - Without even looking at (A, C) we can reason that (A, C) is *Contradiction*.



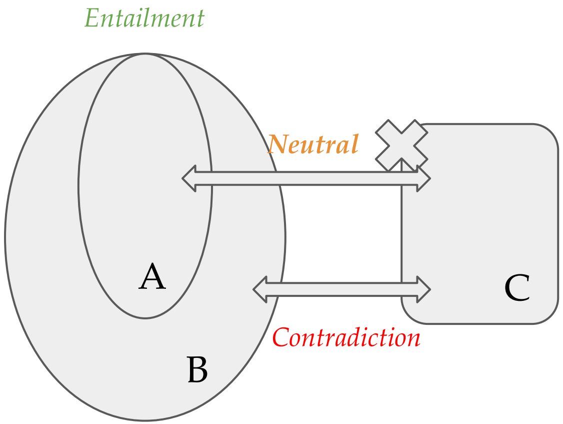
Example: Human Reasoning

- Consider the sentences:
  - A: *Bob is **on a train** to Berlin*
  - B: *Bob is **traveling** to Berlin*
  - C: *Bob is **eating** in Berlin*
- We know -
  - (A, B): *Entailment*
  - (B, C): *Contradiction*
  - Without even looking at (A, C) we can reason that (A, C) is *Contradiction*.
- Unfortunately -
  - Some models may think otherwise.



When the model fails ...

- Consider the sentences:
  - A: *Bob is **on a train** to Berlin*
  - B: *Bob is **traveling** to Berlin*
  - C: *Bob is **eating** in Berlin*

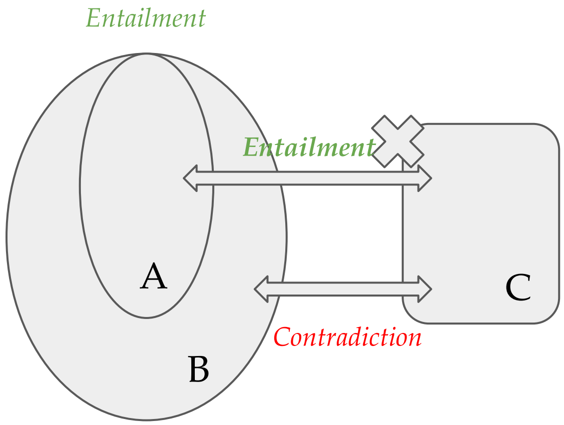


	Input	True	Entailment	Contradiction	Neutral
`decomposable-attention-elmo-2020.04.09`	(A, B)	E	**0.796**	0.020	0.184
	(B, C)	C	0.361	**0.371**	0.268
	(A, C)	C	0.267	0.273	**0.461**

Decomposable attention model with ELMo: https://arxiv.org/abs/1606.01933

… even the SOTA

- Consider the sentences:
  - A: *Bob is **traveling** to Berlin*
  - B: *Bob is **on his way** to Berlin*
  - C: *Bob is **having dinner** in Berlin*

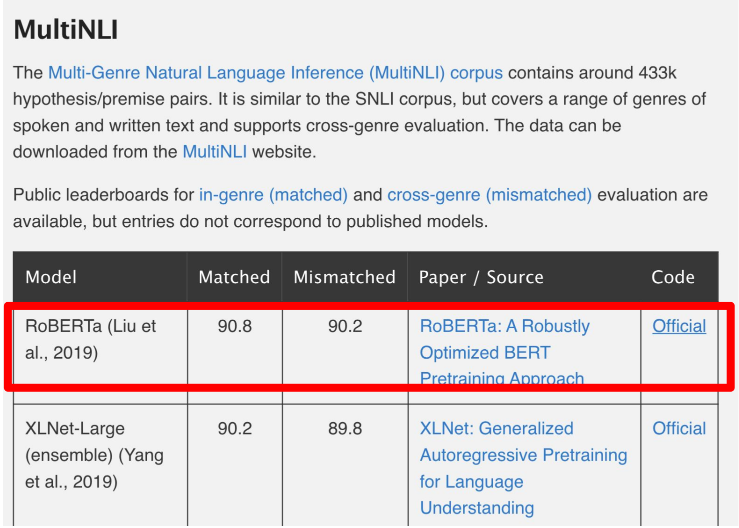


	Input	True	Entailment	Contradiction	Neutral
mnli_roberta-2020.06.09	(A, B)	E	**0.993**	0.000	0.006
	(B, C)	C	0.206	**0.492**	0.302
	(A, C)	C	**0.814**	0.020	0.166

RoBERTa model: https://arxiv.org/abs/1907.11692

… even the SOTA

- Consider the sentences:
  - A: *Bob is **traveling** to Berlin*
  - B: *Bob is **on his way** to Berlin*
  - C: *Bob is **having dinner** in Berlin*



MultiNLI

The Multi-Genre Natural Language Inference (MultiNLI) corpus contains around 433k hypothesis/premise pairs. It is similar to the SNLI corpus, but covers a range of genres of spoken and written text and supports cross-genre evaluation. The data can be downloaded from the MultiNLI website.

Public leaderboards for in-genre (matched) and cross-genre (mismatched) evaluation are available, but entries do not correspond to published models.

Model	Matched	Mismatched	Paper / Source	Code
RoBERTa (Liu et al., 2019)	90.8	90.2	RoBERTa: A Robustly Optimized BERT Pretraining Approach	Official
XLNet-Large (ensemble) (Yang et al., 2019)	90.2	89.8	XLNet: Generalized Autoregressive Pretraining for Language Understanding	Official

	Input	True	Entailment	Contradiction	Neutral
`mnli_roberta-2020.06.09`	(A, B)	E	**0.993**	0.000	0.006
	(B, C)	C	0.206	**0.492**	0.302
	(A, C)	C	**0.814**	0.020	**0.**166

Leaderboards: https://nlpprogress.com/english/natural_language_inference.html
AllenNLP Text Entailment Demo: https://demo.allennlp.org/textual-entailment/roberta-mnli

Challenge: Consistency

- A good system: draw correct inference *and* be consistent in its beliefs.

Challenge: Consistency

- A good system: draw correct inference *and* be consistent in its beliefs.
- Even highly accurate models can be inconsistent.

Challenge: Consistency

- A good system: draw correct inference *and* be consistent in its beliefs.
- Even highly accurate models can be inconsistent.
- Possible mitigation strategies?

Challenge: Consistency

- A good system: draw correct inference *and* be consistent in its beliefs
- Even highly accurate models can be inconsistent
- Possible mitigation strategies?
  - Include those “adversarial inputs” when train the model.

Challenge: Consistency

- A good system: draw correct inference *and* be consistent in its beliefs
- Even highly accurate models can be inconsistent
- Possible mitigation strategies?
    - Include those "adversarial inputs" when train the model.
        - Annotation is costly; automatically generate labels - computational cost.
        - Models may fail to see the consistency inside - thus failing at testing time.

Challenge: Consistency

- A good system: draw correct inference *and* be consistent in its beliefs
- Even highly accurate models can be inconsistent
- Possible mitigation strategies?
  - Include those “adversarial inputs” when train the model
    - Annotation is costly; automatically generate labels - computational cost.
    - Models may fail to see the consistency inside - thus failing at testing time.
  - **Incorporate knowledge** in training

Challenge: Consistency

- A good system: draw correct inference *and* be consistent in its beliefs
- Even highly accurate models can be inconsistent
- Possible mitigation strategies?
  - Include those “adversarial inputs” when train the model
    - Annotation is costly; automatically generate labels - computational cost.
    - Models may fail to see the consistency inside - thus failing at testing time.
  - **Incorporate knowledge** in training (*this paper*).
    - Q: Can we incorporate knowledge in *post-processing*?

Challenge: Consistency

- A good system: draw correct inference *and* be consistent in its beliefs
- Even highly accurate models can be inconsistent
- Possible mitigation strategies?
  - Include those "adversarial inputs" when train the model
    - Annotation is costly; automatically generate labels - computational cost.
    - Models may fail to see the consistency inside - thus failing at testing time.
  - **Incorporate knowledge** in training (*this paper*).
    - Q: Can we incorporate knowledge in *posi* *oc*

As logic forms.

Knowledge as first-order logic

- Let x be a *collection of examples* (labelled or unlabelled), all constraints can be expressed as

$$\bigwedge_{(L,R)} L(x) \rightarrow R(x)$$

  where L and R are Boolean formulas constructed from model predictions on examples in x.

Knowledge as first-order logic

- Let x be a *collection of examples* (labelled or unlabelled), all constraints can be expressed as

$$\bigwedge_{(L,R)} L(x) \rightarrow R(x)$$

Equivalently $L(x) \vee \neg R(x)$

where L and R are Boolean formulas constructed from model predictions on examples in x.

Knowledge as first-order logic

- Let x be a *collection of examples* (labelled or unlabelled), all constraints can be expressed as

$$\bigwedge_{(L,R)} L(x) \rightarrow R(x)$$

Equivalently $L(x) \vee \neg R(x)$

where L and R are Boolean formulas constructed from model predictions on examples in x.

- The rule used in the running example can be written as

$$\mathrm{E}(A, B) \wedge \mathrm{C}(B, C) \rightarrow \mathrm{C}(A, C).$$

Knowledge as first-order logic

- Let x be a *collection of examples* (labelled or unlabelled), all constraints can be expressed as

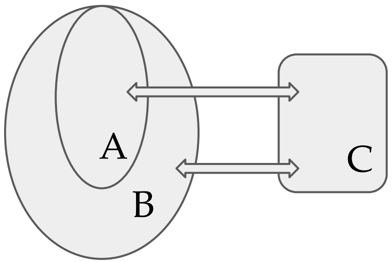
$$\bigwedge_{(L,R)} L(x) \rightarrow R(x)$$

Equivalently $L(x) \vee \neg R(x)$

where L and R are Boolean formulas constructed from model predictions on examples in x.

- The rule used in the running example can be written as

$$\mathrm{E}(A, B) \wedge \mathrm{C}(B, C) \rightarrow \mathrm{C}(A, C).$$



Knowledge as first-order logic

- Let x be a *collection of examples* (labelled or unlabelled), all constraints can be expressed as

$$\bigwedge_{(L,R)} L(x) \rightarrow R(x)$$

Equivalently $L(x) \vee \neg R(x)$

where L and R are Boolean formulas constructed from model predictions on examples in x.

- The rule used in the running example can be written as

$$\mathrm{E}(A, B) \wedge \mathrm{C}(B, C) \rightarrow \mathrm{C}(A, C).$$

- The rule that "model should predict x_i's true label y_i on x_i" can be written as

Knowledge as first-order logic

- Let x be a *collection of examples* (labelled or unlabelled), all constraints can be expressed as

$$\bigwedge_{(L,R)} L(x) \to R(x)$$

Equivalently $L(x) \vee \neg R(x)$

where L and R are Boolean formulas constructed from model predictions on examples in x.

- The rule used in the running example can be written as

$$\mathrm{E}(A, B) \wedge \mathrm{C}(B, C) \to \mathrm{C}(A, C).$$

- The rule that "model should predict x_i's true label y_i on x_i" can be written as

$$\mathsf{TRUE} \longrightarrow y_i(x_i).$$

Knowledge as first-order logic

- Let x be a *collection of examples* (labelled or unlabelled), all constraints can be expressed as

$$\bigwedge_{(L,R)} L(x) \rightarrow R(x)$$

where L and R are Boolean formulas constructed from model predictions on examples in x.

- The rule used in the running example can be written as

$$\mathrm{E}(A, B) \wedge \mathrm{C}(B, C) \rightarrow \mathrm{C}(A, C).$$

This is a constraint on accuracy and will link to the CE loss.

- The rule that "model should predict x_i's true label y_i on x_i" can be written as

$$\mathsf{TRUE} \rightarrow y_i(x_i).$$

Consistency Rules

Annotation Consistency: $\forall\ (P, H), Y \in D, \texttt{TRUE} \rightarrow Y(P, H).$

Symmetry Consistency: $\bigwedge_{(P,H) \in D} C(P, H) \leftrightarrow C(H, P).$

Transitivity Consistency:

$$\begin{aligned}
&\forall (P, H, Z) \in D, \\
&\quad (E\,(P, H) \wedge E\,(H, Z) \rightarrow E\,(P, Z)) \\
&\wedge (E\,(P, H) \wedge C\,(H, Z) \rightarrow C\,(P, Z)) \\
&\wedge (N\,(P, H) \wedge E\,(H, Z) \rightarrow \neg C\,(P, Z)) \\
&\wedge (N\,(P, H) \wedge C\,(H, Z) \rightarrow \neg E\,(P, Z))
\end{aligned}$$

Consistency Rules

If the data is annotated, its label should be predicted.

Annotation Consistency: $\forall\ (P, H), Y \in D, \mathsf{TRUE} \longrightarrow Y(P, H).$

If A contradicts with B, then B also contradicts with A.

Symmetry Consistency: $\bigwedge_{(P,H)\in D} C(P,H) \leftrightarrow C(H,P).$

Transitivity Consistency: $\forall (P,H,Z) \in D,$

$$
\begin{aligned}
&(E(P,H) \wedge E(H,Z) \rightarrow E(P,Z)) \\
\wedge\ &(E(P,H) \wedge C(H,Z) \rightarrow C(P,Z)) \\
\wedge\ &(N(P,H) \wedge E(H,Z) \rightarrow \neg C(P,Z)) \\
\wedge\ &(N(P,H) \wedge C(H,Z) \rightarrow \neg E(P,Z))
\end{aligned}
$$

The running example.

Losses from softened logic

- Predicted label probability (e.g., e(A, B)) as surrogates for Boolean decision (E(A, B)).

Losses from softened logic

- Predicted label probability (e.g., e(*A*, *B*)) as surrogates for Boolean decision (E(*A*, *B*)).
- Transform logic rules to losses by softening them using "*t*-norms".

Name	Boolean Logic	Product	Gödel	Łukasiewicz
Negation	$\neg A$	$1 - a$	$1 - a$	$1 - a$
T-norm	$A \wedge B$	ab	$\min(a, b)$	$\max(0, a + b - 1)$
T-conorm	$A \vee B$	$a + b - ab$	$\max(a, b)$	$\min(1, a + b)$
Residuum	$A \rightarrow B$	$\min\left(1, \frac{b}{a}\right)$	$\begin{cases} 1, & \text{if b} \geq \text{a}, \\ b, & \text{else} \end{cases}$	$\min(1, 1 - a + b)$

Losses from softened logic

- Predicted label probability (e.g., e(A, B)) as surrogates for Boolean decision (E(A, B)).
- Transform logic rules to losses by softening them using "t-norms".

Name	Boolean Logic	Product	Gödel	Łukasiewicz
Negation	$\neg A$	$1 - a$	$1 - a$	$1 - a$
T-norm	$A \wedge B$	ab	$\min(a, b)$	$\max(0, a + b - 1)$
T-conorm	$A \vee B$	$a + b - ab$	$\max(a, b)$	$\min(1, a + b)$
Residuum	$A \rightarrow B$	$\min\left(1, \frac{b}{a}\right)$	$\begin{cases} 1, & \text{if b} \geq \text{a}, \\ b, & \text{else} \end{cases}$	$\min(1, 1 - a + b)$

Losses from softened logic

- Predicted label probability (e.g., e(*A*, *B*)) as surrogates for Boolean decision (E(*A*, *B*)).
- Transform logic rules to losses by softening them using "*t*-norms".

Name	Boolean Logic	Product	Gödel	Łukasiewicz
Negation	$\neg A$	$1 - a$		$1 - a$
T-norm	$A \wedge B$	ab	$\min(a, b)$	$\max(0, a + b - 1)$
T-conorm	$A \vee B$	$a + b - ab$	$\max(a, b)$	$\min(1, a + b)$
Residuum	$A \rightarrow B$	$\min\left(1, \frac{b}{a}\right)$	$\begin{cases} 1, & \text{if b} \geq \text{a}, \\ b, & \text{else} \end{cases}$	$\min(1, 1 - a + b)$

1 - (1-*a*)(1-*b*)

$z \wedge x \leq y \Leftrightarrow z \leq (x \rightarrow y)$

Residuum of Product *t*-norm: https://wikipedia.org/wiki/T-norm#Residuum

Losses from softened logic

Annotation Consistency $\forall\ (P, H), Y \in D, \texttt{TRUE} \rightarrow Y(P, H).$

Symmetry Consistency

$$\bigwedge_{(P,H)\in D} C(P, H) \leftrightarrow C(H, P).$$

Transitivity Consistency

$$\begin{aligned}
&\forall (P, H, Z) \in D, \\
&\quad (E(P, H) \wedge E(H, Z) \rightarrow E(P, Z)) \\
&\quad \wedge (E(P, H) \wedge C(H, Z) \rightarrow C(P, Z)) \\
&\quad \wedge (N(P, H) \wedge E(H, Z) \rightarrow \neg C(P, Z)) \\
&\quad \wedge (N(P, H) \wedge C(H, Z) \rightarrow \neg E(P, Z))
\end{aligned}$$

Losses from softened logic

Annotation Consistency $\forall\ (P, H), Y \in D$, TRUE $\longrightarrow Y(P, H)$.

$$\prod_{(P,H),Y^\star \in D} y^\star_{(P,H)}$$

Annotation Loss L_{ann}

Symmetry Consistency

$$\bigwedge_{(P,H)\in D} C(P,H) \leftrightarrow C(H,P).$$

$$L_{sym} = \sum_{(P,H)\in D} |\log c_{(P,H)} - \log c_{(H,P)}|$$

Symmetry Loss L_{sym}

Transitivity Consistency

$$\begin{aligned} &\forall (P,H,Z) \in D, \\ &\quad (E(P,H) \wedge E(H,Z) \rightarrow E(P,Z)) \\ &\quad \wedge (E(P,H) \wedge C(H,Z) \rightarrow C(P,Z)) \\ &\quad \wedge (N(P,H) \wedge E(H,Z) \rightarrow \neg C(P,Z)) \\ &\quad \wedge (N(P,H) \wedge C(H,Z) \rightarrow \neg E(P,Z)) \end{aligned}$$

$$\begin{aligned} &\mathrm{ReLU}(\log e(P,H) + \log e(H,Z) - \log e(P,Z)) \\ +&\mathrm{ReLU}(\log e(P,H) + \log c(H,Z) - \log c(P,Z)) \\ +&\mathrm{ReLU}(\log n(P,H) + \log e(H,Z) - \log(1-c(P,Z))) \\ +&\mathrm{ReLU}(\log n(P,H) + \log c(H,Z) - \log(1-e(P,Z))) \end{aligned}$$

Transitivity Loss L_{tran}

Losses from softened logic

Annotation Consistency $\forall\ (P, H), Y \in D$, TRUE $\rightarrow Y(P, H)$.

$$\prod_{(P,H),Y^\star \in D} y^\star_{(P,H)}$$

Annotation Loss L_{ann}

Sy

1. Easy to optimize!
2. Utilize *both* labelled and un-labelled data!

(F

$$L_{sym} = \sum_{(P,H)\in D} |\log c_{(P,H)} - \log c_{(H,P)}|$$

Symmetry Loss L_{sym}

Tr

$$\begin{aligned}\forall (P, H, Z) \in D,\\ &(E(P,H) \wedge E(H,Z) \rightarrow E(P,Z))\\ \wedge &(E(P,H) \wedge C(H,Z) \rightarrow C(P,Z))\\ \wedge &(N(P,H) \wedge E(H,Z) \rightarrow \neg C(P,Z))\\ \wedge &(N(P,H) \wedge C(H,Z) \rightarrow \neg E(P,Z))\end{aligned}$$

$$\begin{aligned}&\mathrm{ReLU}(\log e(P,H) + \log e(H,Z) - \log e(P,Z))\\ +&\mathrm{ReLU}(\log e(P,H) + \log c(H,Z) - \log c(P,Z))\\ +&\mathrm{ReLU}(\log n(P,H) + \log e(H,Z) - \log(1 - c(P,Z)))\\ +&\mathrm{ReLU}(\log n(P,H) + \log c(H,Z) - \log(1 - e(P,Z)))\end{aligned}$$

Transitivity Loss L_{tran}

Training models, with knowledge

- The loss $L = L_{ann} + \lambda_{sym} L_{sym} + \lambda_{tran} L_{tran}$ can be minimized via off-the-shelf optimizers.
- Symmetry and transitivity losses does *not* require labelled data!
- Training
  - BERT/LSTM bases fined tuned on SNLI/MultiNLI then fined tuned again.
  - SNLI/MultiNLI: labelled data for *annotation consistency*.
  - **Mirrored (M):** Swap two sentences in labelled examples, for *symmetry consistency*.
  - **Unlabelled Triples (T):** Sample triples from COCO dataset for *transitivity consistency*.
  - **Unlabelled Pairs (U):** Swap the first pair in each triple in (T) for *symmetry consistency*.
- Testing
  - Sample new sets using the same procedure for evaluation.

MS COCO Dataset: https://arxiv.org/abs/1405.0312

Measuring inconsistencies

- ❏ Global Violation (ϱ)

$$\rho = \frac{\sum_{x \in D} \left[\bigvee_{(L,R)} \neg \left(L(x) \rightarrow R(x) \right) \right]}{|D|}$$

- ❏ Conditional Violation (τ)

$$\tau = \frac{\sum_{x \in D} \left[\bigvee_{(L,R)} \neg \left(L(x) \rightarrow R(x) \right) \right]}{\sum_{x \in D} \left[\bigvee_{(L,R)} L(x) \right]}$$

Measuring inconsistencies

- Global Violation (ϱ)

# of violations

$$\rho = \frac{\sum_{x \in D} \left[\bigvee_{(L,R)} \neg (L(x) \rightarrow R(x)) \right]}{|D|}$$

Dataset size

Indicator function

- Conditional Violation (τ)

$$\tau = \frac{\sum_{x \in D} \left[\bigvee_{(L,R)} \neg (L(x) \rightarrow R(x)) \right]}{\sum_{x \in D} \left[\bigvee_{(L,R)} L(x) \right]}$$

Measuring inconsistencies

- Global Violation (ϱ)

# of violations

Indicator function

$$\rho = \frac{\sum_{x \in D} \left[\bigvee_{(L,R)} \neg (L(x) \rightarrow R(x)) \right]}{|D|}$$

Dataset size

If a model tries *not* to satisfy any L, ρ can also be quite low!

- Conditional Violation (τ)

$$\tau = \frac{\sum_{x \in D} \left[\bigvee_{(L,R)} \neg (L(x) \rightarrow R(x)) \right]}{\sum_{x \in D} \left[\bigvee_{(L,R)} L(x) \right]}$$

$L \rightarrow R$ is a tautology if L is FALSE. Here τ *only* considers those with a true L.

Experiment Insights

Highly accurate models may also be very inconsistent.

Training set used

Base Model

Fine tuned on

Config	5%				100%			
	ρ_S	τ_S	ρ_T	τ_T	ρ_S	τ_S	ρ_T	τ_T
BERT w/ SNLI	26.3	64.4	4.9	14.8	18.6	60.3	4.7	14.9
BERT w/ MultiNLI	28.4	69.3	7.0	18.5	20.6	58.9	5.6	17.5
BERT w/ SNLI+MultiNLI	25.3	62.4	4.8	14.8	18.1	59.6	4.5	14.8
BERT w/ SNLI+MultiNLI[2]	22.1	67.1	4.1	13.7	19.3	59.7	4.5	15.2
LSTM w/ SNLI+MultiNLI	25.8	69.5	9.9	21.0	16.8	53.6	5.3	16.0

If predicts (A, B) as Contradiction, then at least 60% chance it predicts (B, A) as something else.

ρ_S : *Global symmetry inconsistency*
τ_S : *Conditional symmetry inconsistency*
ρ_T : *Global transitivity inconsistency*
τ_T : *Conditional transitivity inconsistency*

$$\bigwedge_{(P,H)\in D} C(P,H) \leftrightarrow C(H,P).$$

$$\begin{aligned}\forall(P,H,Z) \in D,\\ &(E(P,H) \wedge E(H,Z) \rightarrow E(P,Z))\\ \wedge&(E(P,H) \wedge C(H,Z) \rightarrow C(P,Z))\\ \wedge&(N(P,H) \wedge E(H,Z) \rightarrow \neg C(P,Z))\\ \wedge&(N(P,H) \wedge C(H,Z) \rightarrow \neg E(P,Z))\end{aligned}$$

Experiment Insights

Highly accurate models may also be very inconsistent.

Training set used

Base Model / *Fine tuned on*

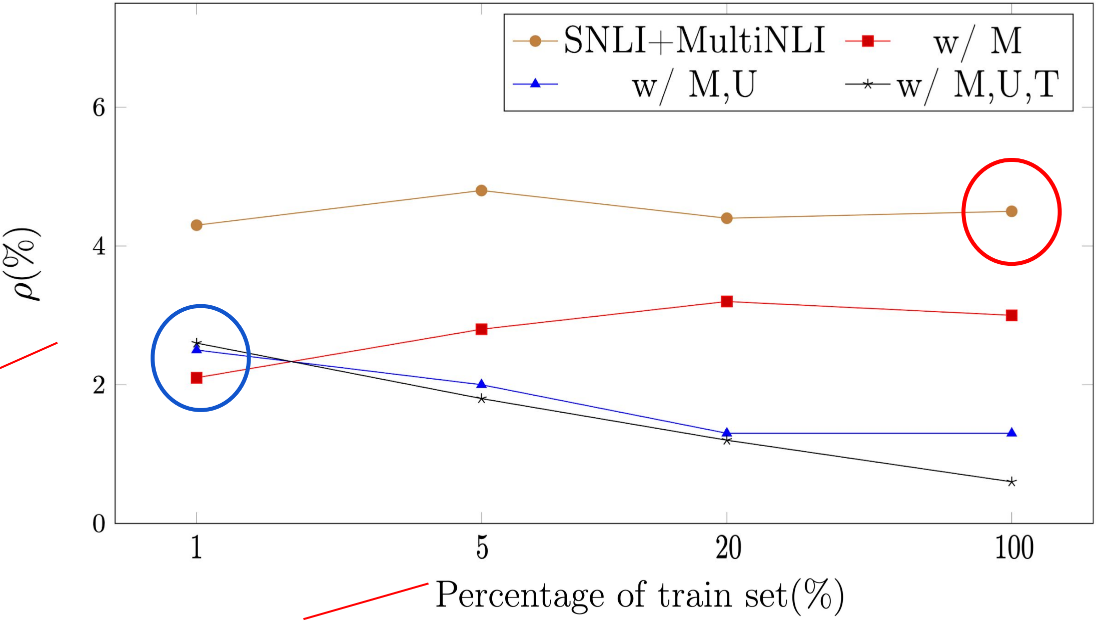
Config	5%				100%			
	ρ_S	τ_S	ρ_T	τ_T	ρ_S	τ_S	ρ_T	τ_T
BERT w/ SNLI	26.3	64.4	4.9	14.8	18.6	60.3	4.7	14.9
BERT w/ MultiNLI	28.4	69.3	7.0	18.5	20.6	58.9	5.6	17.5
BERT w/ SNLI+MultiNLI	25.3	62.4	4.8	14.8	18.1	59.6	4.5	14.8
BERT w/ SNLI+MultiNLI[2]	22.1	67.1	4.1	13.7	19.3	59.7	4.5	15.2
LSTM w/ SNLI+MultiNLI	25.8	69.5	9.9	21.0	16.8	53.6	5.3	16.0

If predicts (A, B) as Contradiction, then at least 60% chance it predicts (B, A) as something else.

ρ_S : *Global symmetry inconsistency*
τ_S : *Conditional symmetry inconsistency*
ρ_T : *Global transitivity inconsistency*
τ_T : *Conditional transitivity inconsistency*

Experiment Insights

Weak supervision with constraints trained models can be more consistent than a baseline trained on the full dataset.



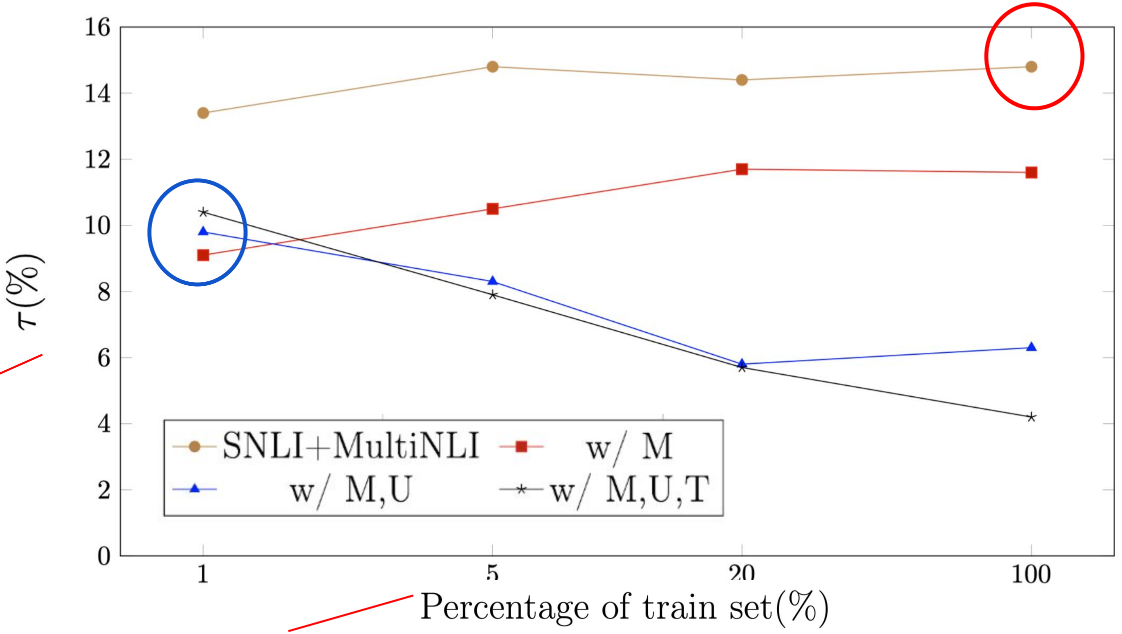
Global transitivity inconsistency (smaller is better)

Amount of label supervision

(M) *Symmetry Consistency (SLI datasets)*
(U) *Symmetry Consistency (COCO)*
(T) *Transitivity Consistency (COCO)*

Experiment Insights

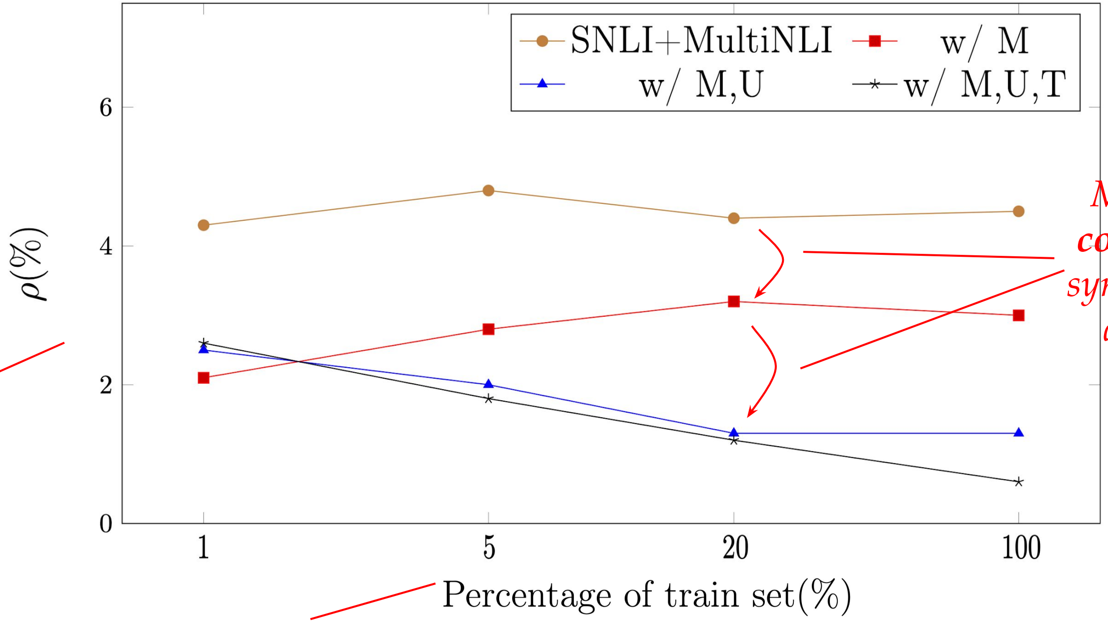
Weak supervision with constraints trained models can be ***more consistent*** *than a baseline trained on the full dataset.*



(M) *Symmetry Consistency (SLI datasets)*
(U) *Symmetry Consistency (COCO)*
(T) *Transitivity Consistency (COCO)*

Experiment Insights

*Constraints do not conflict with each other. They are **mutually beneficial**.*



*More **transitively consistent** when the symmetry consistency datasets are used.*

Global transitivity inconsistency (smaller is better)

Amount of label supervision

Experiment Insights

Constraints does not reduce model accuracy.

Datasets the BERT base model trained on

Training set used

Test datasets to evaluate test accuracy on

Config	1%		5%		20%		100%	
	SNLI	MultiNLI	SNLI	MultiNLI	SNLI	MultiNLI	SNLI	MultiNLI
SNLI+MultiNLI	79.7	70.1	84.6	77.2	87.8	80.6	90.1	83.5
SNLI+MultiNLI^2	80.3	71.0	85.3	77.4	87.9	80.7	90.3	84.0
w/ M	80.1	71.0	85.3	77.8	88.1	80.6	90.3	84.1
w/ M,U	80.2	71.0	85.4	77.2	88.1	80.9	90.5	84.3
w/ M,U,T	80.6	71.1	85.4	77.2	88.1	80.9	90.2	84.2

Prediction accuracy not dropped (even increased) when more constraints are enforced.

(M) *Symmetry Consistency (SLI datasets)*
(U) *Symmetry Consistency (COCO)*
(T) *Transitivity Consistency (COCO)*

Experiment Insights

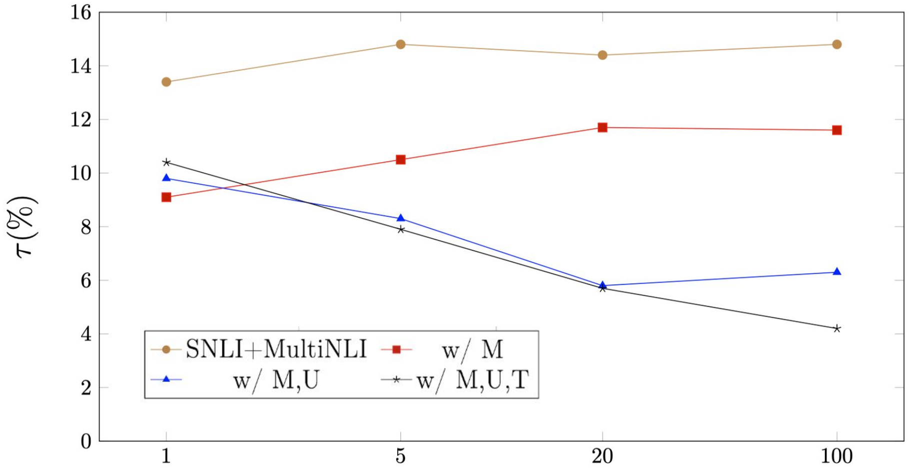
- ❑ Highly accurate models may also be very inconsistent.
- ❑ Weak supervision with constraints is already very helpful for consistency.
- ❑ Constraints do *not* conflict with each other. They are *mutually beneficial*.
- ❑ Adding more constraints does *not* reduce model accuracy.

Summary

- Knowledge as supervision: use logic rules to guide learning with consistency constraints.
  - *What* is learnt here?
  - What *generalization* is supported?

Summary

- Knowledge as supervision: use logic rules to guide learning with consistency constraints.
  - *What* is learnt here?
  - What *generalization* is supported?



Summary

- ❏ Knowledge as supervision: use logic rules to guide learning with consistency constraints.
  - ❏ *What* is learnt here?
  - ❏ What *generalization* is supported?
- ❏ Many consistency constraints do not require annotated data - enabling utilization of both labelled and un-labelled data.
- ❏ The constraints were shown to be mutually beneficial and do not hinder prediction accuracy.

Summary

- Source of knowledge: *human knowledge (logic)*.
- How to learn models: *fine-tuning on datasets made for enforcing consistencies*.
- Encoding constraints: *softened logic*.
- Global inference: *hard*.

Comments

- Constraints only applied in training time. No mechanism to enforce them during test time.

Comments

- Constraints only applied in training time. No mechanism to enforce them during test time.

- A good system: draw correct inference *and* be consistent in its beliefs.

Comments

Since the constraints are over multiple instances.

- Constraints only applied in training time. No mechanism to enforce them during test time.
- A good system: draw correct inference *and* be consistent in its beliefs.

What if its beliefs are wrong?

Discussions

- Is it possible to automatically discover and enforce consistency constraints?

Discussions

- Is it possible to automatically discover and enforce consistency constraints?
- Also used in many recent work, though the industry has long been using methods such as ILP for enforcing constraints.
  - Designing consistent QA system by augment labelled training data;

Discussions

- Is it possible to automatically discover and enforce consistency constraints?
- Also used in many recent work, though the industry has long been using methods such as ILP for enforcing constraints.
  - Designing consistent QA system by augment labelled training data;

Logic-Guided Data Augmentation and Regularization for Consistent Question Answering

Akari Asai† and **Hannaneh Hajishirzi**†‡
†University of Washington ‡Allen Institute for AI
{akari, hannaneh}@cs.washington.edu

[AH20] https://arxiv.org/abs/2004.10157

Discussions

- ❏ Is it possible to automatically discover and enforce consistency constraints?
- ❏ Also used in many recent work, though the industry has long been using methods such as ILP for enforcing constraints.
    - ❏ Improve semantic role labelling models, improvements under low-resource scenarios;

Discussions

- Is it possible to automatically discover and enforce consistency constraints?
- Also used in many recent work, though the industry has long been using methods such as ILP for enforcing constraints.
    - Improve semantic role labelling models, improvements under low-resource scenarios;

Structured Tuning for Semantic Role Labeling

Tao Li
University of Utah
tli@cs.utah.edu

Parth Anand Jawale
University of Colorado
Parth.Jawale@colorado.edu

Martha Palmer
University of Colorado
Martha.Palmer@colorado.edu

Vivek Srikumar
University of Utah
svivek@cs.utah.edu

[LJPS20] https://arxiv.org/abs/2005.00496

Discussions

- Is it possible to automatically discover and enforce consistency constraints?
- Also used in many recent work, though the industry has long been using methods such as ILP for enforcing constraints.
  - Event-event relation extraction with low jointly labelled data.

Discussions

- ❑ Is it possible to automatically discover and enforce consistency constraints?
- ❑ Also used in many recent work, though the industry has long been using methods such as ILP for enforcing constraints.
  - ❑ Event-event relation extraction with low jointly labelled data.

Joint Constrained Learning for Event-Event Relation Extraction

Haoyu Wang[1], **Muhao Chen**[1], **Hongming Zhang**[2*] **& Dan Roth**[1]
[1]Department of Computer and Information Science, UPenn
[2]Department of Computer Science and Engineering, HKUST
{why16gzl, muhao, danroth}@seas.upenn.edu; hzhangal@cse.ust.hk

[WCZR20] https://arxiv.org/abs/2010.06727

Discussions

- Is it possible to automatically discover and enforce consistency constraints?
- Also used in many recent work, though the industry has long been using methods such as ILP for enforcing constraints.
    - Designing consistent QA system by augment labelled training data;
    - Improve semantic role labelling models, improvements under low-resource scenarios;
    - Event-event relation extraction with low jointly labelled data.

Discussions

- (Vivian, Matthew): *inconsistency decreased, but accuracy not changed much. Why?*
- (Yahan): *more data are labelled as "Neutral" after consistency-constrained training. Why?*
- (Dan): *approximate knowledge (softend logic) used here - perhaps not optimizing the best objective. Need to find way to prevent "shifts" and to prevent correctly predicted sampled to be predicted wrong due to consistency constraints. Perhaps add new constraints.*

References

- Tao Li, Vivek Gupta, Maitrey Mehta, and Vivek Srikumar. 2019. *A logic-driven framework for consistency of neural models*. EMNLP.
- Haoyu Wang, Muhao Chen, Hongming Zhang, and Dan Roth. 2020. Joint constrained learning for event-event relation extraction. EMNLP.
- Asai, Akari, and Hannaneh Hajishirzi. 2020. Logic-guided data augmentation and regularization for consistent question answering. arXiv:2004.10157.
- Tao Li, Parth Anand Jawale, Martha Palmer, and Vivek Srikumar. 2020. Structured tuning for semantic role labeling. arXiv:2005.00496.