A Logic-Driven Framework for Consistency of Neural Models

Tao Li, Vivek Gupta, Maitrey Mehta, Vivek Srikumar

THE UNIVERSITY OF UTAH

Key Points

Neural models are good at individual predictions. But they can be ***inconsistent*** across examples. To model (in)consistencies, we present:

1. A mechanism to measure output inconsistency w.r.t. to declaratively specified invariants.
2. A framework that compiles domain knowledge stated in first-order logic to loss components, mitigating inconsistency.
3. An analysis of the impact of consistencies and predictive accuracy, showing that accuracy alone is not sufficient.

Errors & Metrics

Suppose we have 3 sentences:

***P**: John is on a train to Berlin.*
***H**: John is traveling to Berlin.*
***Z**: John is having lunch in Berlin.*

In NLI, we know that **P** entails **H** and **H** contradicts **Z**.

What about **P** and **Z** ? We can write a simple rule:

*if **P** entails **H**, and **H** contradicts **Z**,*
*then **P** contradicts **Z**.*

Generally, we can express such domain knowledge in first-order logic:

$$\forall x \in D, \quad \bigwedge_{(L,R)} L(x) \rightarrow R(x)$$

where x: a collection of examples.

To measure errors, we define two metrics:

Global violation rate ρ

$$\frac{\text{\#instances with violation}}{\text{\#instances}} = \frac{\sum_{x\in D}\left[\bigvee_{(L,R)} \neg(L(x)\rightarrow R(x))\right]}{|D|}$$

Conditional violation rate τ

$$\frac{\text{\#instances with violation}}{\text{\#instances where LHS holds}} = \frac{\sum_{x\in D}\left[\bigvee_{(L,R)} \neg(L(x)\rightarrow R(x))\right]}{\sum_{x\in D}\left[\bigvee_{(L,R)} L(x)\right]}$$

Case Study: NLI

Annotation Consistency *(i.e. Accuracy)*

model prediction should agree with annotation.

$\forall (P,H), Y^\star \in D, \quad \top \rightarrow Y^\star(P,H)$

where $Y^\star$: the ground truth label.

Mirror Consistency

P contradicts H iff. H also contradicts P.

$\forall (P,H) \in D, \quad C(P,H) \leftrightarrow C(H,P)$

where C : the *Contradiction* label.

A BERT model has τ≈60% violation while random guess has τ≈67% !!

Transitivity Consistency

label transitivity with any sentence triple.

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

where E : the *Entailment* label, N : the *Neutral* label.

Relaxing Logic

The question is how to incorporate these non-differentiable rules in an end-to-end training framework.

Triangular norm (t-norm) defines a systematic way to relax logic. We use the ***product*** t-norm.

Logic	Relaxation
$\neg A$	$1-a$
$A \wedge B$	ab
$A \rightarrow B$	$\min\left(1, \frac{b}{a}\right)$

We want the invariants to be true. Equivalently, we want their relaxations to be maximally true.

1. Consistencies apply to all examples, forming a huge conjunction, which becomes summation in log space.
2. Particularly, the *Annotation* consistency becomes cross-entropy loss.

Labeled & Unlabeled Data

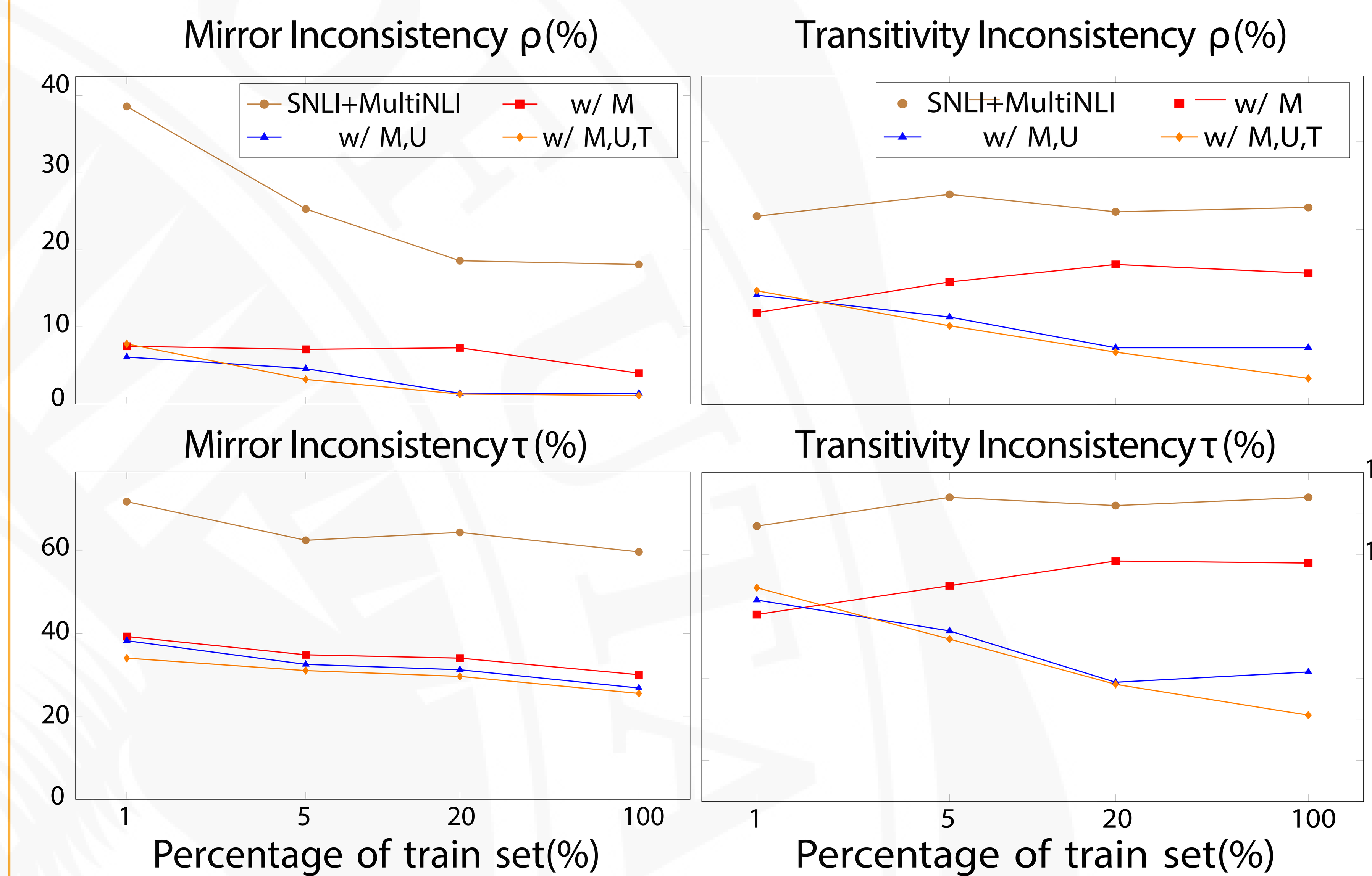
We use SNLI, MultiNLI, and MSCOCO captions.

	Labeled	Mirror labeled (M)	Mirror unlabeled (U)	Transitivity (T)
SNLI	✅	✅		
MultiNLI	✅	✅		
MSCOCO			✅	✅

From MSCOCO, we sampled 100k unlabeled sentence triples for training, and another 100k for evaluation.

Experiments

We use the derived losses to finetune BERT base.



With our inconsistency losses, the BERT models become significantly more consistent. Meanwhile, the accuracies (i.e. annotation consistency) remain on par [-0.2, +0.2] across different settings.

With 100% labeled data, BERT model has 90+ accuracy but terrible consistencies. With 1% labeled data, our framework yields more consistent models than training unconstrainedly with full data.

*i.e. **Accuracy and consistency are complementary metrics**.*

Training with mirror consistency does not guarantee better transitivity consistency (the red curve above).

Conclusions

1. Our framework introduces a general way to design loss functions using the ***product*** t-norm ( ).
2. No extra trainable parameters are required.
3. Models are both accurate and consistent at the same time.
4. Standard evaluations focus on accuracy but not on the mirror/transitivity consistencies.

For details, please refer to our paper.

Thanks for stopping by!