Analysis methods in NLP: Overview

Christopher Potts

Stanford Linguistics

CS224u: Natural language understanding

Overview

Behavioral evaluations

- Adversarial testing
- Adversarial training and testing

Structural evaluation methods

- Probing
- Feature attribution

Motivations

1. Finding the limits of you system
2. Understanding system behavior
3. Achieving more robust systems

The techniques we discuss are powerful and easy ways to improve the analysis section of a final paper!

The story of an adversarial test

	Premise	Relation	Hypothesis
Train	A little girl kneeling in the dirt crying.	entails	A little girl is very **sad**.
Adversarial		entails	A little girl is very **unhappy**.
Train	An elderly couple are sitting outside a restaurant, enjoying wine.	entails	A couple drinking **wine**.
Adversarial		neutral	A couple drinking **champagne**.

Glockner et al. 2018

The story of an adversarial test

Model	Train set	SNLI test set	New test set	Δ
Decomposable Attention (Parikh et al., 2016)	SNLI	84.7%	51.9%	-32.8
	MultiNLI + SNLI	84.9%	65.8%	-19.1
	SciTail + SNLI	85.0%	49.0%	-36.0
ESIM (Chen et al., 2017)	SNLI	87.9%	65.6%	-22.3
	MultiNLI + SNLI	86.3%	74.9%	-11.4
	SciTail + SNLI	88.3%	67.7%	-20.6
Residual-Stacked-Encoder (Nie and Bansal, 2017)	SNLI	86.0%	62.2%	-23.8
	MultiNLI + SNLI	84.6%	68.2%	-16.8
	SciTail + SNLI	85.0%	60.1%	-24.9
WordNet Baseline	-	-	85.8%	-
KIM (Chen et al., 2018)	SNLI	88.6%	83.5%	-5.1

Models that have access to the resources used to create the adversarial examples

Table 3: Accuracy of various models trained on SNLI or a union of SNLI with another dataset (MultiNLI, SciTail), and tested on the original SNLI test set and the new test set.

The story of an adversarial test

RoBERTA-MNLI, off-the-shelf

[1]:
```
import nli, os, torch
from sklearn.metrics import classification_report
```

[2]:
```
# Available from https://github.com/BIU-NLP/Breaking_NLI:
breaking_nli_src_filename = os.path.join("../new-data/data/dataset.jsonl")
reader = nli.NLIReader(breaking_nli_src_filename)
```

[3]:
```
exs = [((ex.sentence1, ex.sentence2), ex.gold_label) for ex in reader.read()]
```

[4]:
```
X_test_str, y_test = zip(*exs)
```

[5]:
```
model = torch.hub.load('pytorch/fairseq', 'roberta.large.mnli')
_ = model.eval()
```

```
Using cache found in /Users/cgpotts/.cache/torch/hub/pytorch_fairseq_master
```

[6]:
```
X_test = [model.encode(*ex) for ex in X_test_str]
```

[7]:
```
pred_indices = [model.predict('mnli', ex).argmax() for ex in X_test]
```

[8]:
```
to_str = {0: 'contradiction', 1: 'neutral', 2: 'entailment'}
```

[9]:
```
preds = [to_str[c.item()] for c in pred_indices]
```

The story of an adversarial test

RoBERTA-MNLI, off-the-shelf

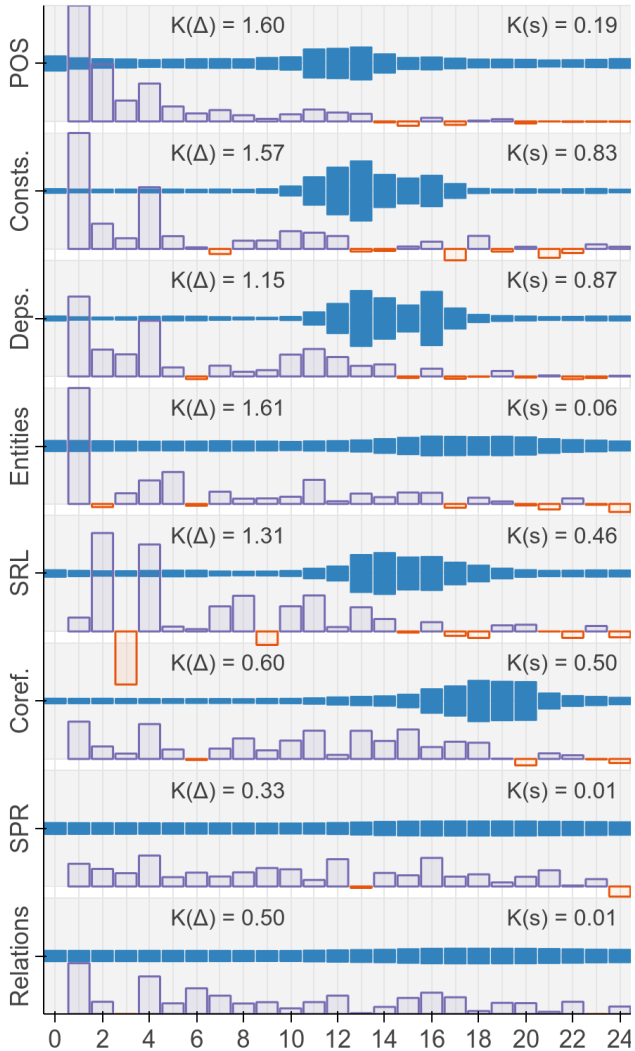
[10]:
```
print(classification_report(y_test, preds))
```

```
                precision    recall  f1-score   support

 contradiction       0.99      0.97      0.98      7164
    entailment       0.86      1.00      0.92       982
       neutral       0.15      0.15      0.15        47

      accuracy                           0.97      8193
     macro avg       0.67      0.71      0.68      8193
  weighted avg       0.97      0.97      0.97      8193
```

Adversarial training (and testing)

1. Commonsense reasoning (Zellers et al. 2018, 2019)
2. NLI (see Nie et al. 2020)
3. QA (see Bartolo et al. 2020)
4. Sentiment (DynaSent; Potts et al. 2020)
5. Hate Speech (Vidgen et al. 2020)

Probing internal representations



Tenney et al. 2019

Feature attribution

True Label	Predicted (Prob)	Word-level importance
positive	positive (0.85)	[CLS] They sell a mean apple pie . [SEP]
positive	positive (0.68)	[CLS] They make a mean apple pie . [SEP]
positive	positive (0.97)	[CLS] He makes a mean apple pie . [SEP]
positive	negative (0.15)	[CLS] He sells a mean apple pie . [SEP]

Integrated gradients; Sundararajan et al. 2017

References I

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the AI: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Christopher Potts, Zhengxuan Wu, Atticus Geiger, and Douwe Kiela. 2020. DynaSent: A dynamic benchmark for sentiment analysis. *arXiv preprint arXiv:2012.15349*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2020. Learning from the worst: Dynamically generated datasets to improve online hate detection. *arXiv prerint arXiv:2012.15761*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.