GAN-CodeBERT

李拙

lizhmq@pku.edu.cn

Outline

- Motivation
- SS-GAN
- GAN-BERT
- GAN-CodeBERT

Motivation

Background

- Semi-supervised learning
  - Suppose we are solving a K classification problem
  - Labeled data $(x, y) \sim D_L$, Unlabeled data $x \sim D_U$
- Can we use unlabeled data to help train the classifier?
- Idea: GAN does not need label
  - “Real” data is the label compared with fake (generated) data

SS-GAN

Semi-supervised GAN[1]

- Discriminator:
  - Discriminate between fake and real examples
  - Learn to classify meanwhile (class $K+1$ is fake)

$$L = -E_{(x,y)\sim D_L} \log\big(p(y|x)\big) - E_{x\in D_U}(\log(1-p(y=K+1|x))) - E_{x\in G}\log(p(y=K+1|x))$$

- Generator:
  - Mislead the discriminator

$$L = -E_{x\in G}\log(1-p(y=K+1|x))$$

Semi-supervised GAN[1]

- Discriminator:
  - Discriminate & Classify

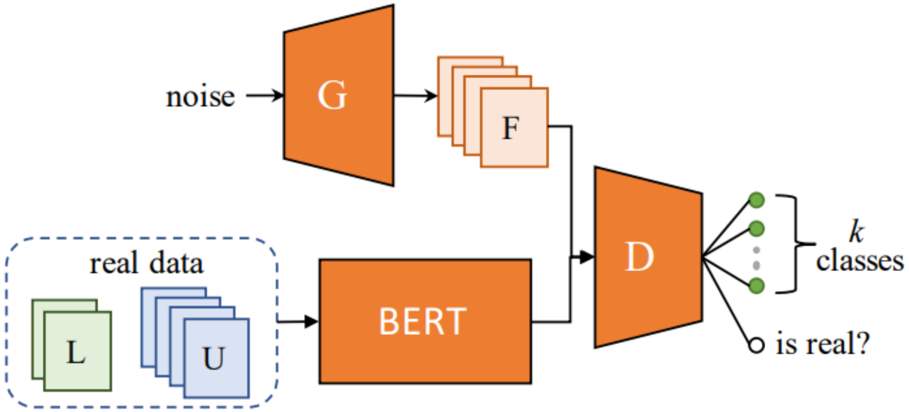
$$L = -E_{(x,y)\sim D_L}\log\big(p(y|x)\big) - E_{x\in D_U}(\log(1-p(y=K+1|x))) - E_{x\in G}\log(p(y=K+1|x))$$

- Generator:
  - Mislead the discriminator
- Target: the classification model
- Compared with supervised learning, discriminator further incorporate the loss of discriminating fake examples
- This adversarial process force discriminator to extract useful feature from labeled and unlabeled real examples

GAN-BERT

GAN-BERT[2]

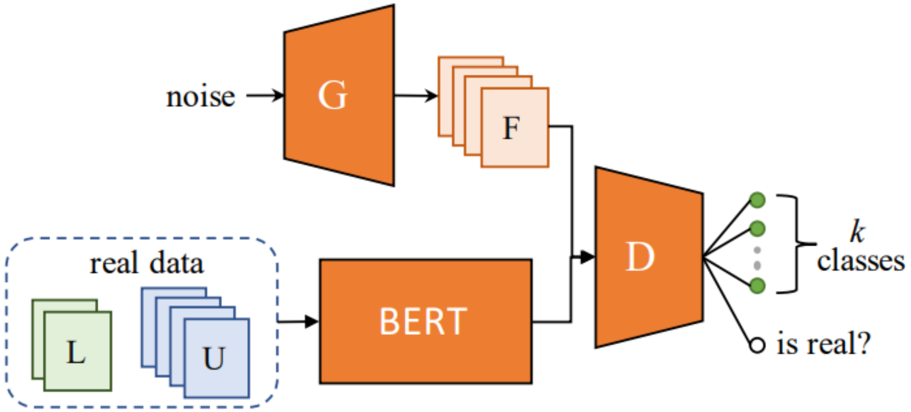
- Difficult to generate high-quality fake sentence
- Then generate fake feature directly
- Generator:
    - Generate 768 dimension representation (fake)
- Discriminator:
    - Discriminate fake rep of G and BERT rep
    - Classify real rep



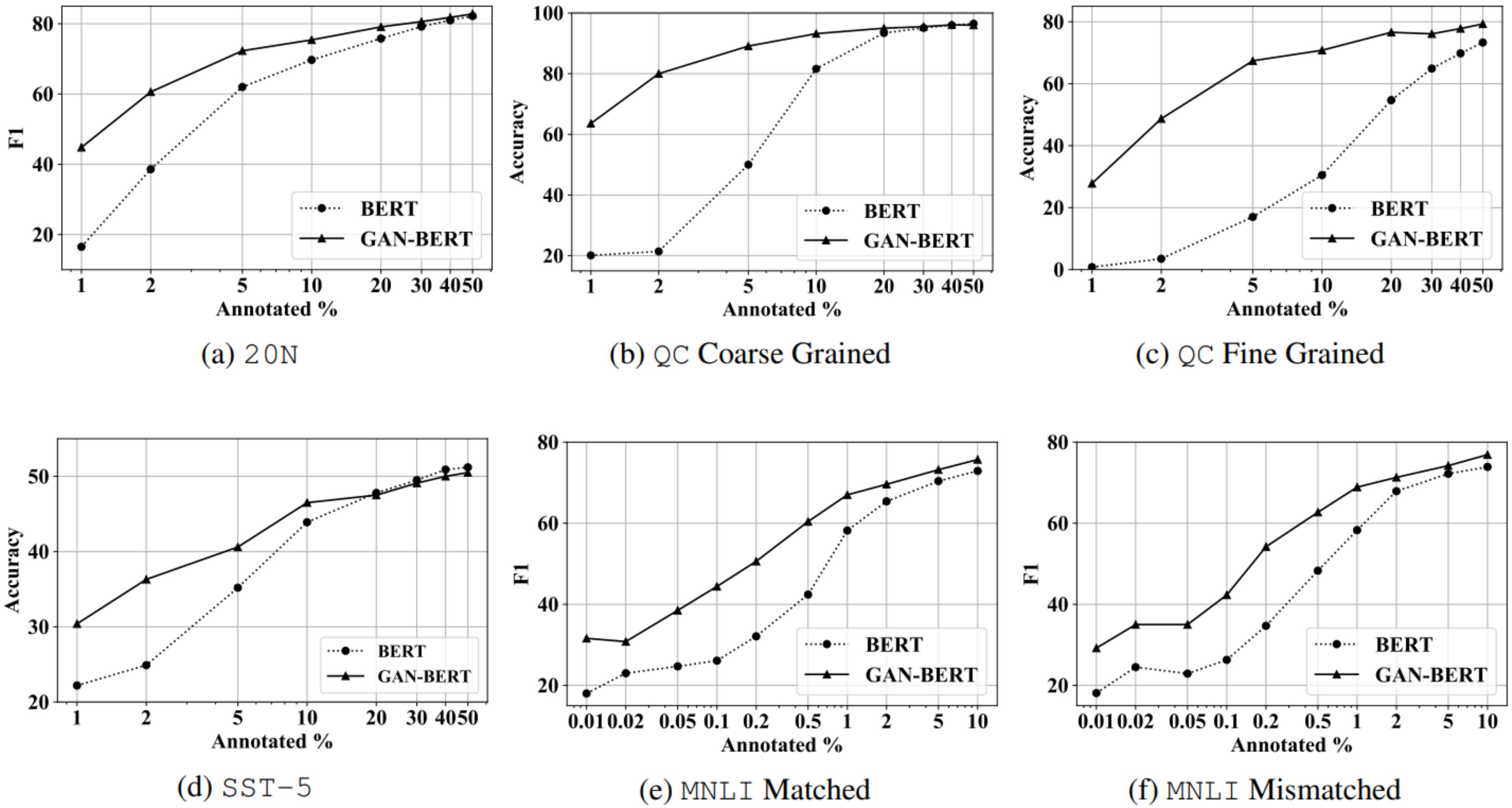
GAN-BERT[2]

- Discriminator loss:

$$\begin{aligned} L = & -E_{(x,y)\sim D_L} log\big(p(y|BERT(x))\big) \\ & -E_{x\in D_U}\big(log\big(1-p(y=K+1|BERT(x))\big)\big) \\ & -E_{h\in G}\, log\big(p(y=K+1|h)\big) \end{aligned}$$



GAN-BERT[2]



(a) 20N

(b) QC Coarse Grained

(c) QC Fine Grained

(d) SST-5

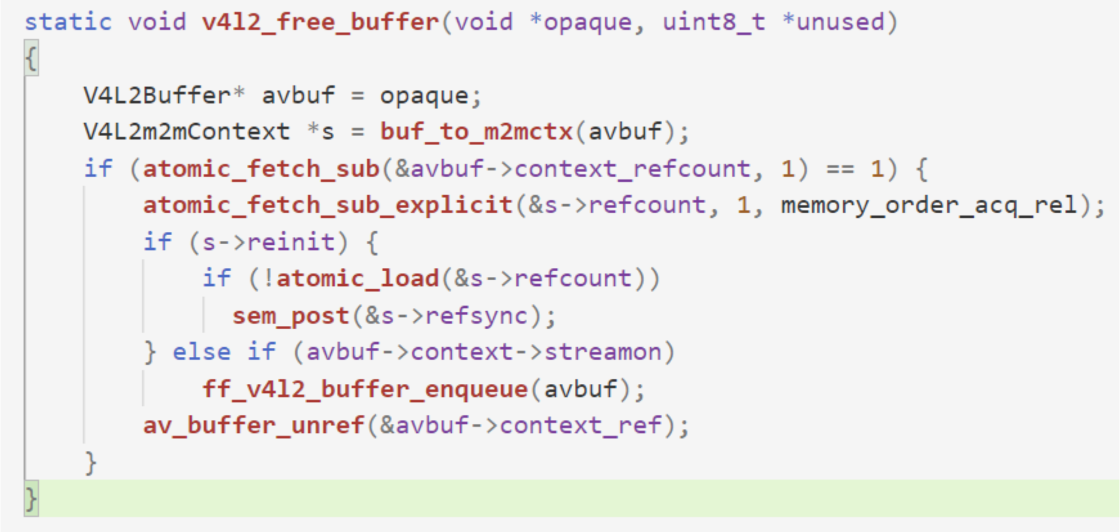
(e) MNLI Matched

(f) MNLI Mismatched

GAN-CodeBERT

GAN-CodeBERT

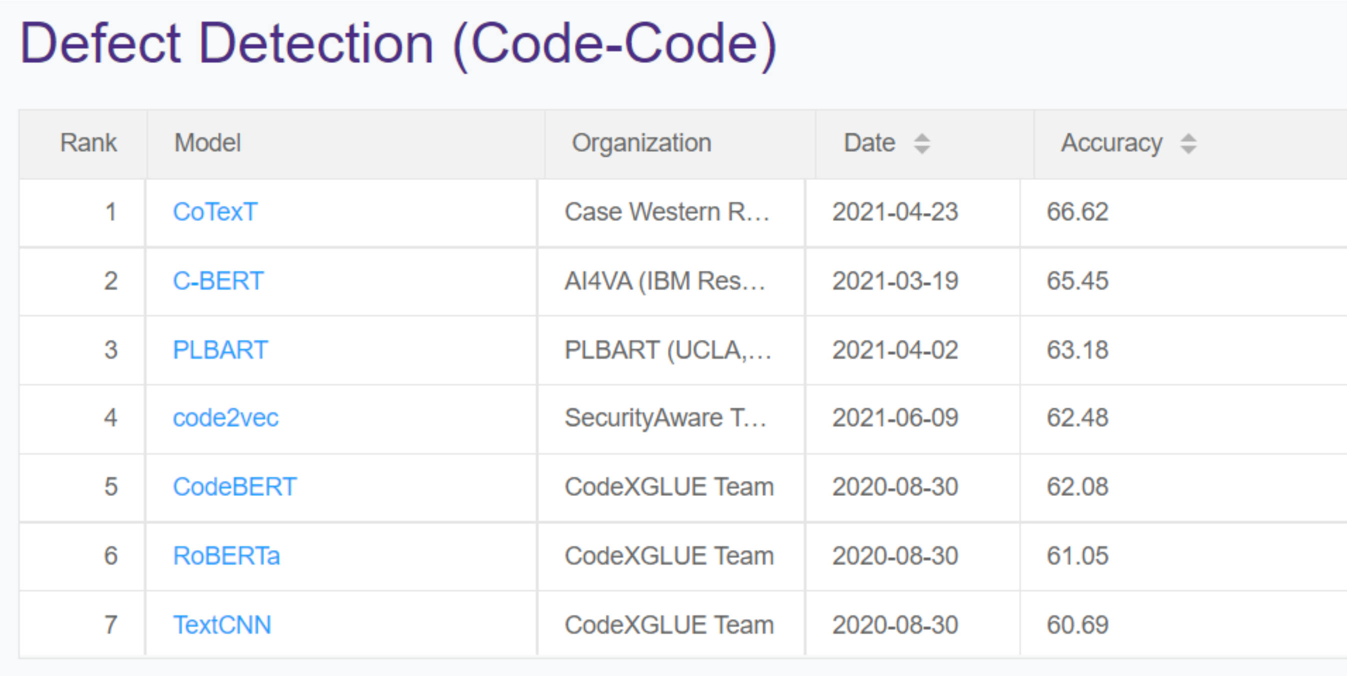
- CodeBERT[3]: BERT-like model pretrained with bimodal NL-Code data
- Task: defect detection
    - To predict whether there is any defects in a piece of code
        - E.g. memory leak
    - Binary classification problem



```
static void v4l2_free_buffer(void *opaque, uint8_t *unused)
{
    V4L2Buffer* avbuf = opaque;
    V4L2m2mContext *s = buf_to_m2mctx(avbuf);
    if (atomic_fetch_sub(&avbuf->context_refcount, 1) == 1) {
        atomic_fetch_sub_explicit(&s->refcount, 1, memory_order_acq_rel);
        if (s->reinit) {
            if (!atomic_load(&s->refcount))
              sem_post(&s->refsync);
        } else if (avbuf->context->streamon)
            ff_v4l2_buffer_enqueue(avbuf);
        av_buffer_unref(&avbuf->context_ref);
    }
}
```

GAN-CodeBERT

- Dataset: Devign[4]
  - 27318 labeled examples
- Leaderboard[5]

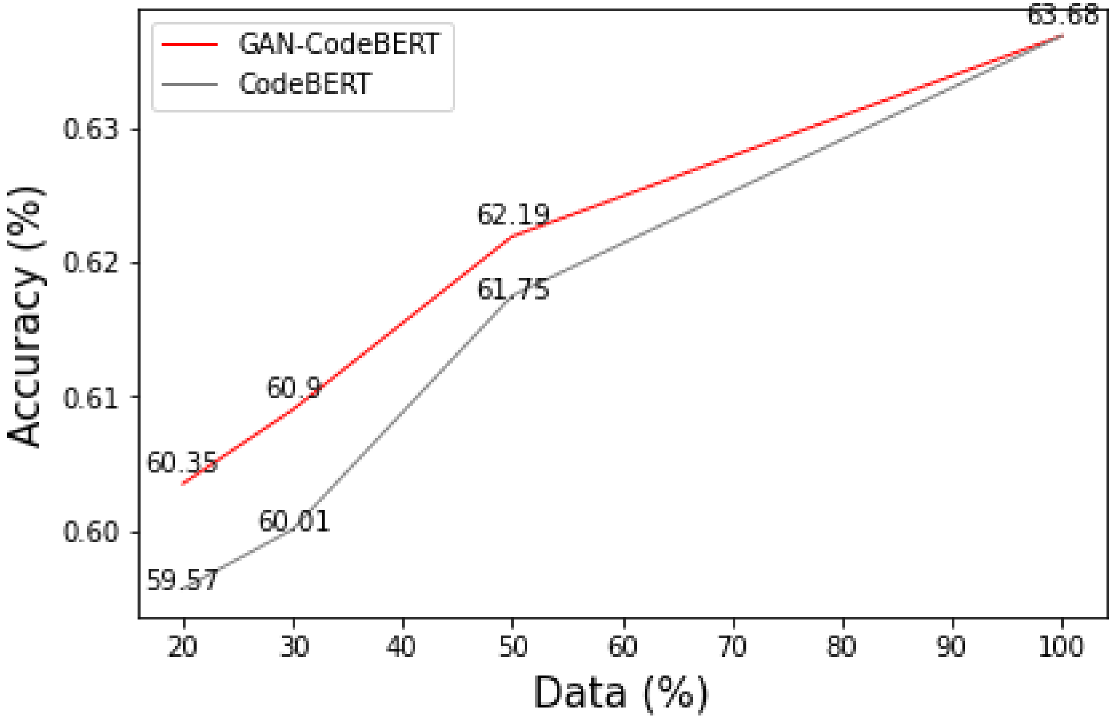


Defect Detection (Code-Code)

Rank	Model	Organization	Date	Accuracy
1	CoTexT	Case Western R…	2021-04-23	66.62
2	C-BERT	AI4VA (IBM Res…	2021-03-19	65.45
3	PLBART	PLBART (UCLA,…	2021-04-02	63.18
4	code2vec	SecurityAware T…	2021-06-09	62.48
5	CodeBERT	CodeXGLUE Team	2020-08-30	62.08
6	RoBERTa	CodeXGLUE Team	2020-08-30	61.05
7	TextCNN	CodeXGLUE Team	2020-08-30	60.69

GAN-CodeBERT

- Experiment results



GAN-CodeBERT

- Open sourced on GitHub:
  - https://github.com/Lizhmq/GAN-CodeBERT

References

- [1] Improved Techniques for Training GANs – NIPs16
- [2] GAN-BERT: Generative Adversarial Learning for Robust Text Classification with a Bunch of Labeled Example – ACL19
- [3] CodeBERT: A Pre-Trained Model for Programming and Natural Languages – EMNLP20
- [4] Devign: Effective vulnerability identification by learning comprehensive program semantics via graph neural networks – NIPs19
- [5] CodeXGLUE: A Machine Learning Benchmark Dataset for Code Understanding and Generation

Thanks.

Q&A