ISC 2018 ISC互联网安全大会 | 360互联网安全中心

Fuzzing on train:
AI制导PDF文件生成技术的探索之旅

邹权臣　　中国信息安全测评中心博士后

马金鑫　　中国信息安全测评中心副研究员

2018 ISC 互联网安全大会 中国 · 北京
Internet Security Conference 2018　Beijing · China
（原中国互联网安全大会）

个人简介

邹权臣

中国信息安全测评中心 博士后

研究方向： 自动化漏洞分析

负责、参与多项国家、省部级科研项目，发表多篇学术论文。

zouquanchen@126.com

马金鑫

中国信息安全测评中心 副研究员

北京邮电大学硕士生导师

研究方向： 软件安全、漏洞分析

主持多项国家、省部级科研项目，发表20余篇学术论文，获得软件著作权5项，发明专利2项，曾发现多个0Day漏洞。

majinxin2003@126.com

目录

研究背景

方案设计

实验分析

结论与展望

研究背景

高结构化样本与PDF文件

AI&样本生成

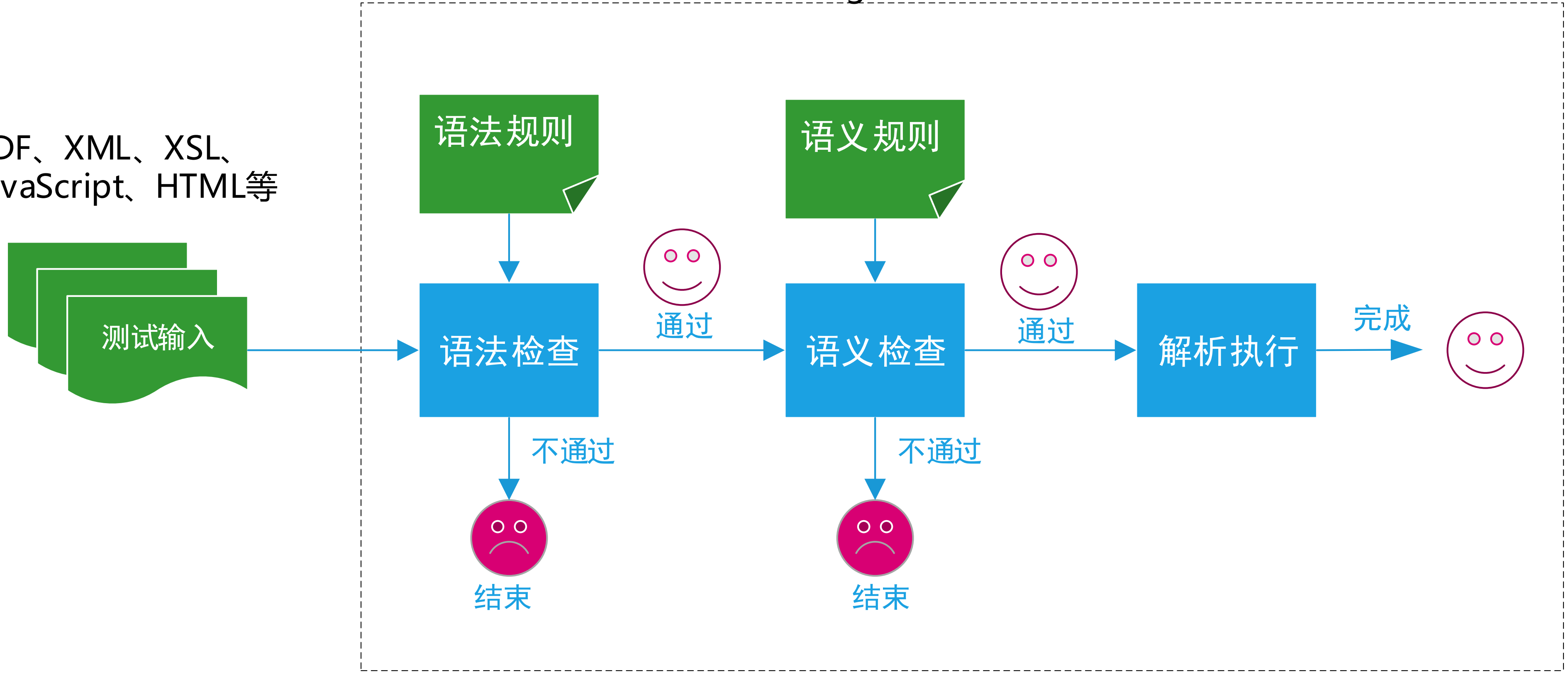
LEARN&FUZZ

高结构化样本与PDF文件

高结构化样本结构复杂，通过规定的语法、语义检查才能被各类解析工具执行

PDF、XML、XSL、
JavaScript、HTML等

Foxit Reader、Adobe Reader、
Chrome、Edge、Firefox等



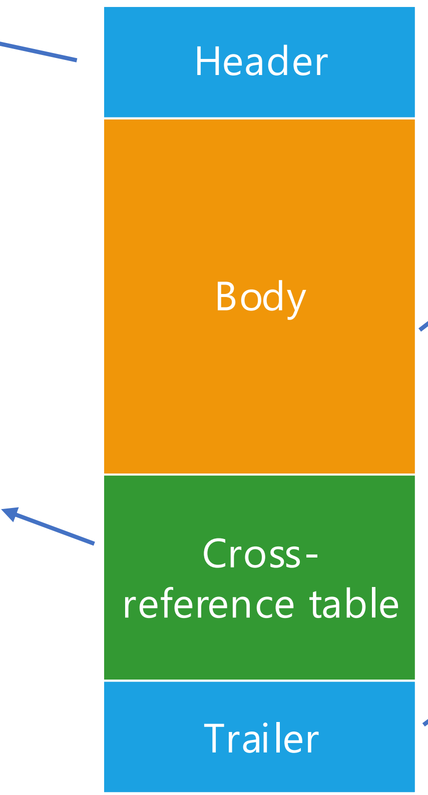
高结构化样本与PDF文件

```
%PDF-1.3
```

PDF文件所遵从的版本号

```
xref
0 257
0000000000 65535 f
0000000017 00000 n
0000000212 00000 n
0000000231 00000 n
0000000251 00000 n
0000000825 00000 n
0000000876 00000 n
```

间接对象地址索引表



由一系列的PDF间接对象组成，如字体、页面、图像等，构成了PDF文件的具体内容（按大类可分为带stream不带stream的obj）

```
8 0 obj
<<
   /Type /Pages
   /Kids[ 22 0 R ]
   /Count 1
>>
endobj
```

```
209 0 obj
<< /Type /XObject /Subtype /Image /Width 51 /Height 69 /BitsPerCom
ponent 8
/ColorSpace 29 0 R /Length 214 /Filter [ /ASCII85Decode /FlateDecode ]
>>
Stream
…
endstream
endobj
```

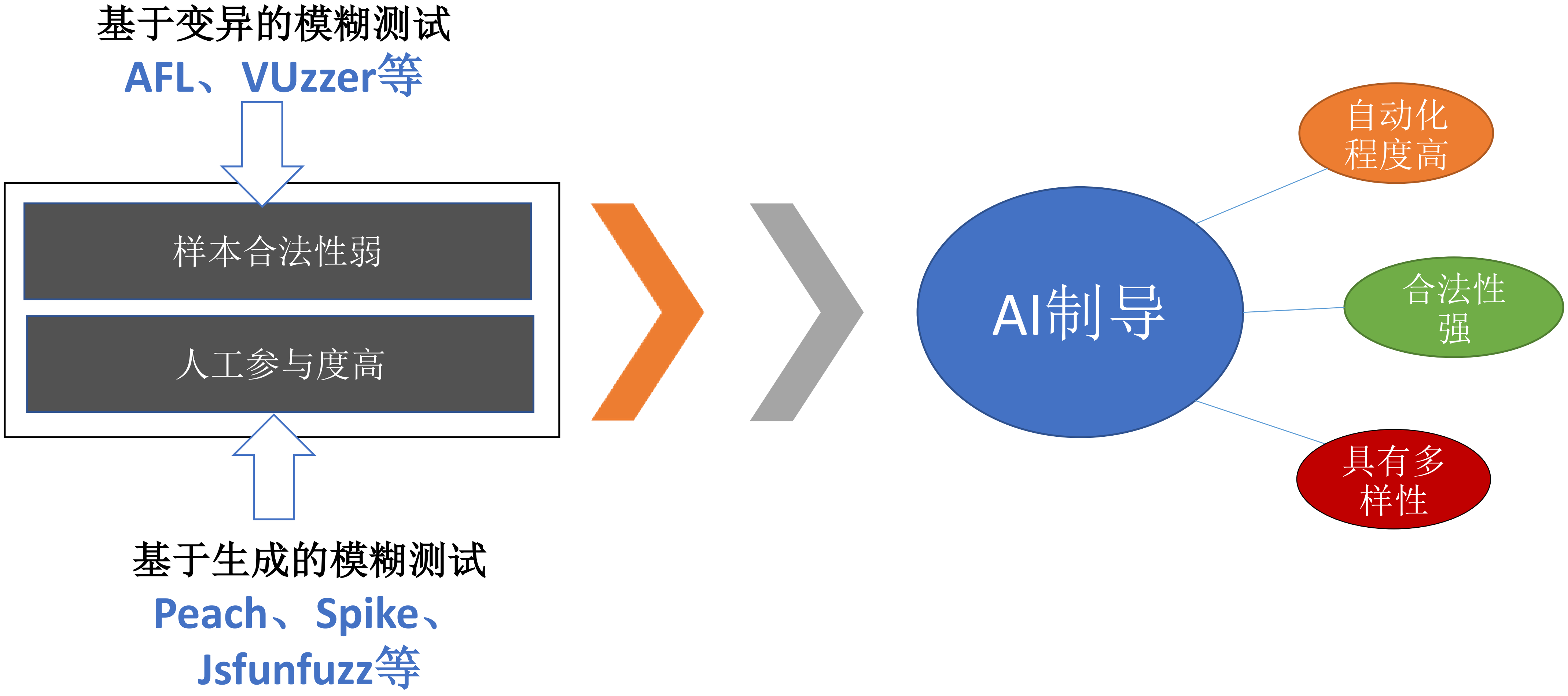
```
trailer
<</Info 19 0 R /Root 21 0 R /Size
257/ID[<15481298DAABCC5184A2001C560B476B><6DC090EE200F6EB
5201096388FFC0D37>]>>
startxref
320283
%%EOF
```

指明根对象(Catalog)，保存了加密等安全信息，并声明交叉引用表的地址

Adobe Systems Incorporated. *PDF Reference*, 6th edition, Nov. 2006. http://www.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/pdf_reference_1-7.pdf

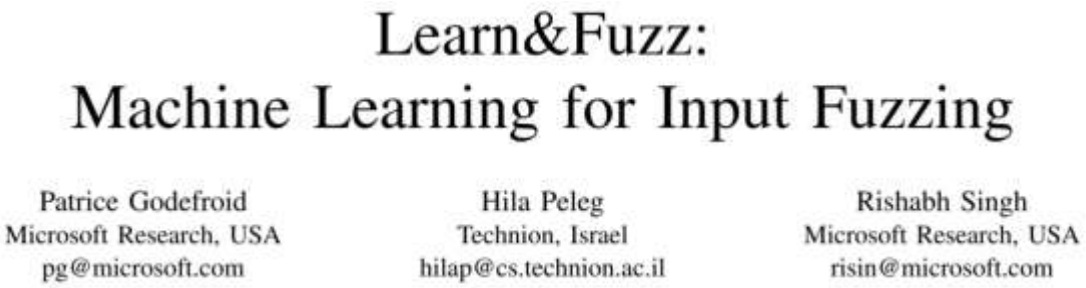
1310页

AI&样本生成



基于变异的模糊测试
AFL、VUzzer等
样本合法性弱
人工参与度高
基于生成的模糊测试
Peach、Spike、
Jsfunfuzz等
AI制导
自动化程度高
合法性强
具有多样性

AI&样本生成



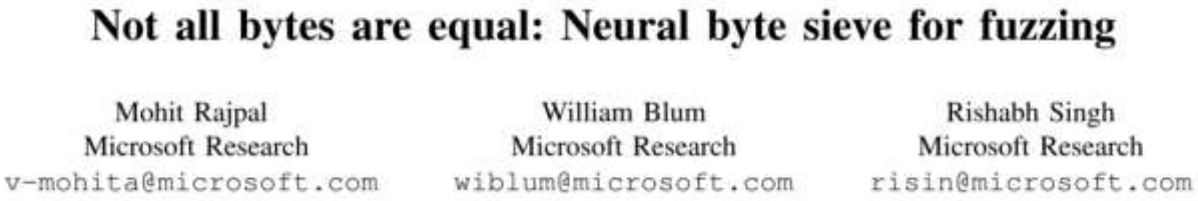
Learn&Fuzz:
Machine Learning for Input Fuzzing

Patrice Godefroid
Microsoft Research, USA
pg@microsoft.com

Hila Peleg
Technion, Israel
hilap@cs.technion.ac.il

Rishabh Singh
Microsoft Research, USA
risin@microsoft.com

ASE’ 17 LSTM



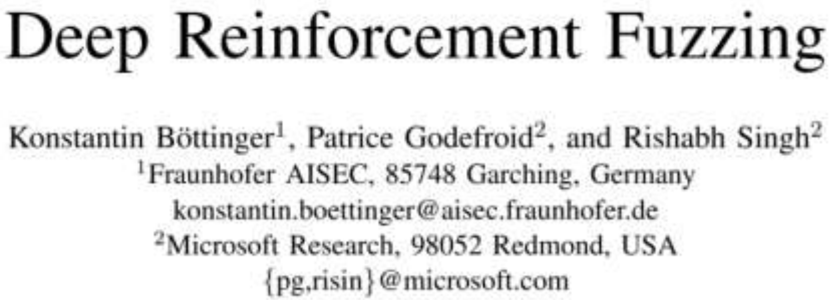
Not all bytes are equal: Neural byte sieve for fuzzing

Mohit Rajpal
Microsoft Research
v-mohita@microsoft.com

William Blum
Microsoft Research
wiblum@microsoft.com

Rishabh Singh
Microsoft Research
risin@microsoft.com

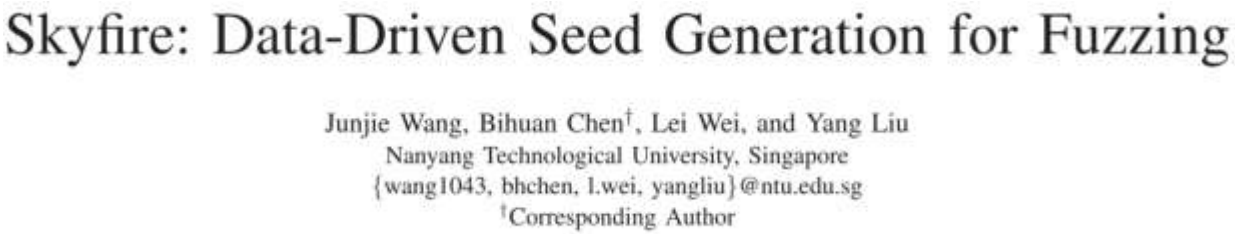
arXiv’ 17 LSTM/BLSTM/seq2seq



Deep Reinforcement Fuzzing

Konstantin Böttinger[1], Patrice Godefroid[2], and Rishabh Singh[2]
[1]Fraunhofer AISEC, 85748 Garching, Germany
konstantin.boettinger@aisec.fraunhofer.de
[2]Microsoft Research, 98052 Redmond, USA
{pg,risin}@microsoft.com

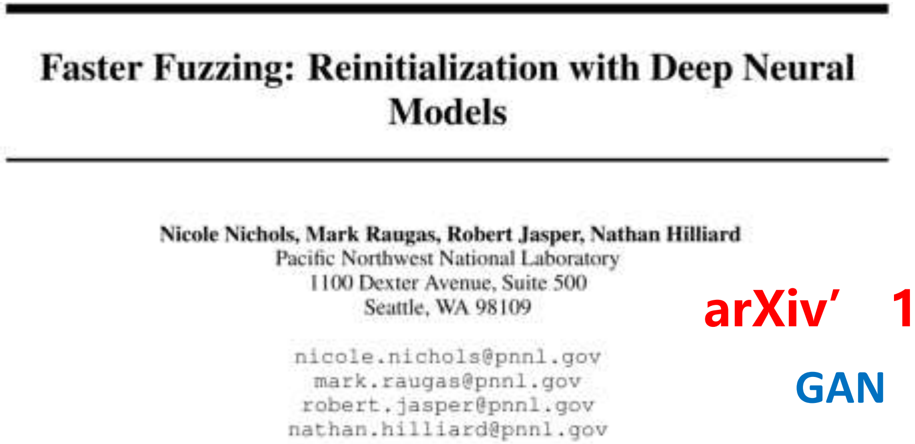
arXiv’ 18 Q-Learning



Skyfire: Data-Driven Seed Generation for Fuzzing

Junjie Wang, Bihuan Chen†, Lei Wei, and Yang Liu
Nanyang Technological University, Singapore
{wang1043, bhchen, l.wei, yangliu}@ntu.edu.sg
†Corresponding Author

S&P’ 17 PCSG



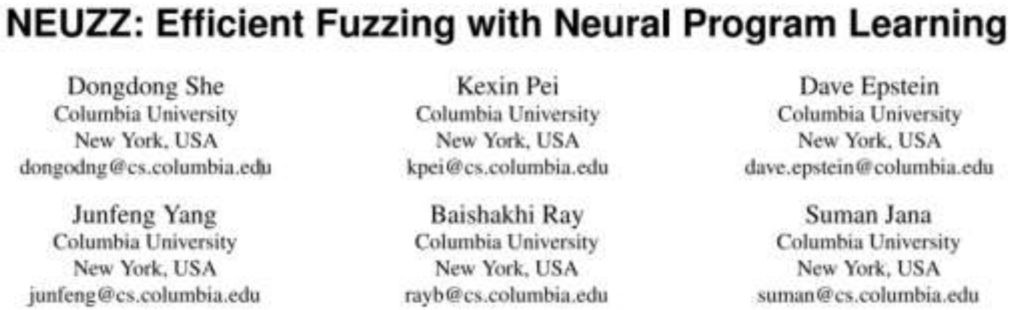
Faster Fuzzing: Reinitialization with Deep Neural Models

Nicole Nichols, Mark Raugas, Robert Jasper, Nathan Hilliard
Pacific Northwest National Laboratory
1100 Dexter Avenue, Suite 500
Seattle, WA 98109

nicole.nichols@pnnl.gov
mark.raugas@pnnl.gov
robert.jasper@pnnl.gov
nathan.hilliard@pnnl.gov

arXiv’ 17

GAN



NEUZZ: Efficient Fuzzing with Neural Program Learning

Dongdong She
Columbia University
New York, USA
dongodng@cs.columbia.edu

Kexin Pei
Columbia University
New York, USA
kpei@cs.columbia.edu

Dave Epstein
Columbia University
New York, USA
dave.epstein@columbia.edu

Junfeng Yang
Columbia University
New York, USA
junfeng@cs.columbia.edu

Baishakhi Ray
Columbia University
New York, USA
rayb@cs.columbia.edu

Suman Jana
Columbia University
New York, USA
suman@cs.columbia.edu

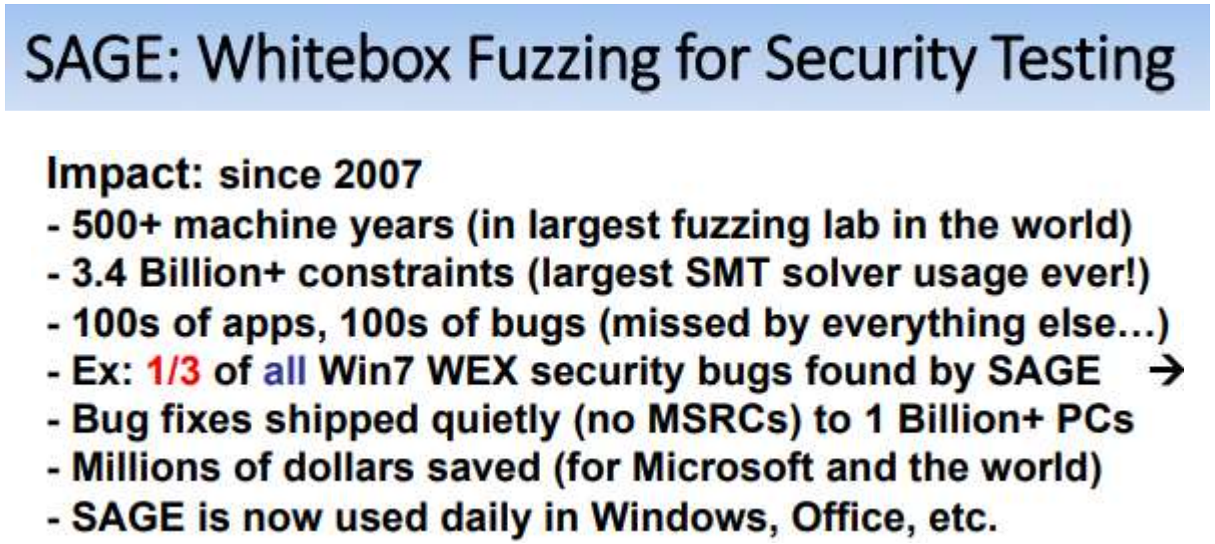
arXiv’ 18 CNN

LEARN&FUZZ

Microsoft Research

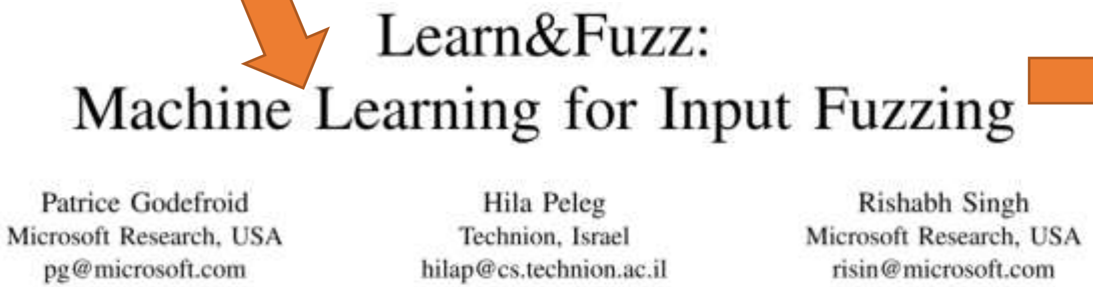
Patrice Godefroid

Email: …g AT microsoft.com
Mail: …rosoft Research, One Microsoft Wa



SAGE: Whitebox Fuzzing for Security Testing

Impact: since 2007
- **500+ machine years (in largest fuzzing lab in the world)**
- **3.4 Billion+ constraints (largest SMT solver usage ever!)**
- **100s of apps, 100s of bugs (missed by everything else…)**
- **Ex: 1/3 of all Win7 WEX security bugs found by SAGE →**
- **Bug fixes shipped quietly (no MSRCs) to 1 Billion+ PCs**
- **Millions of dollars saved (for Microsoft and the world)**
- **SAGE is now used daily in Windows, Office, etc.**



Learn&Fuzz: Machine Learning for Input Fuzzing

Patrice Godefroid
Microsoft Research, USA
pg@microsoft.com

Hila Peleg
Technion, Israel
hilap@cs.technion.ac.il

Rishabh Singh
Microsoft Research, USA
risin@microsoft.com

- **数据集来源：** **Windows fuzzing team**
- **初始测试集：** **63,000** non-binary PDF objects out of **534** PDF files (seed minimization)
- **实验数据集：** **1,000** PDF objects
- **模型：** **LSTM** with **2** hidden layers
- **实验环境：** 4-core 64-bit Windows 10 VMs with 20GB of RAM
- **训练时长：** **50** epoch **10** hours
- **生成PDF数量：** **1,000** per 10 epoch
- **测试结果（Edge）：** **Pass rate(70%-97%)、Instruction coverage、Bugs(1)**

方案设计

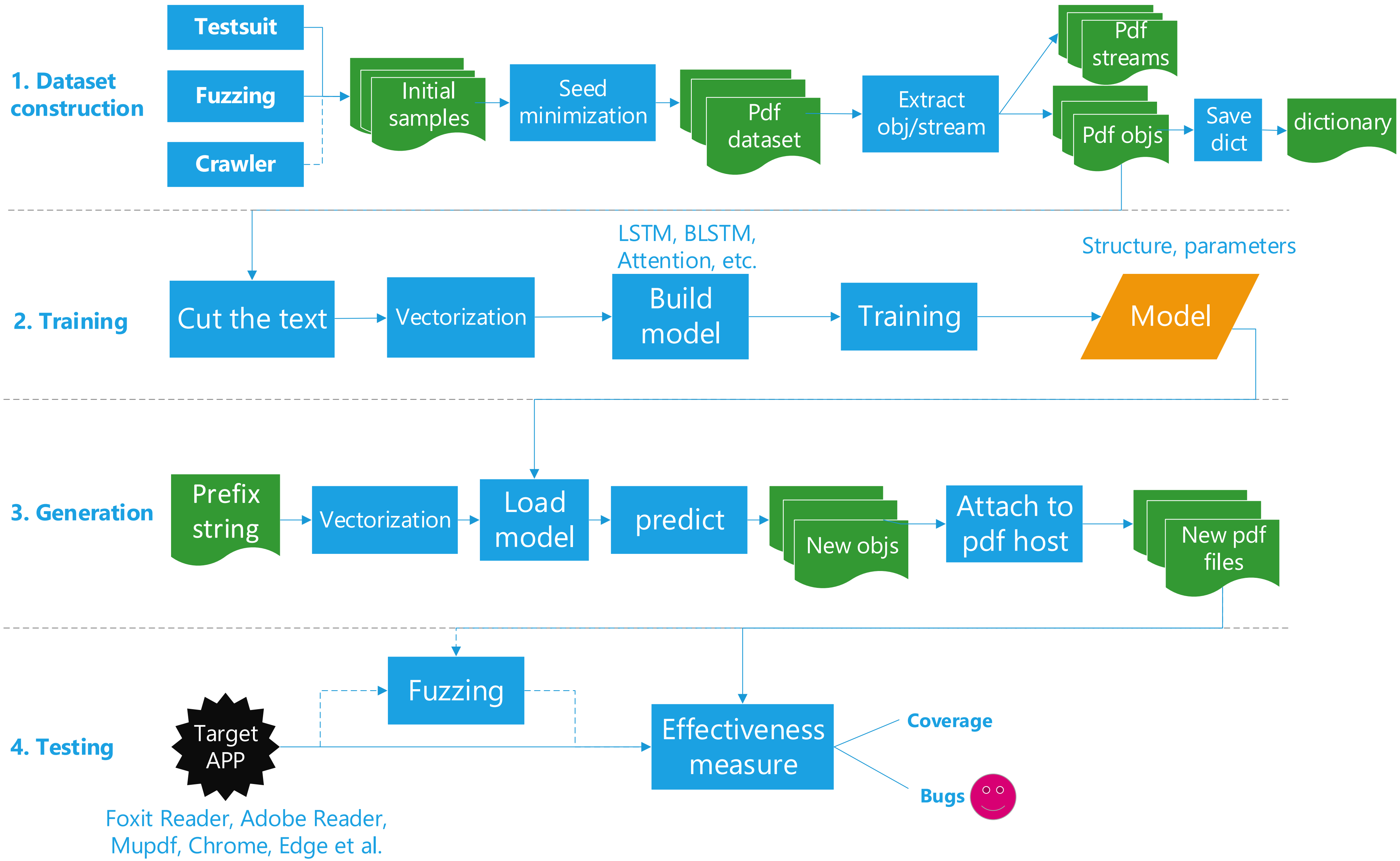
研究方案

数据集构建

模型训练

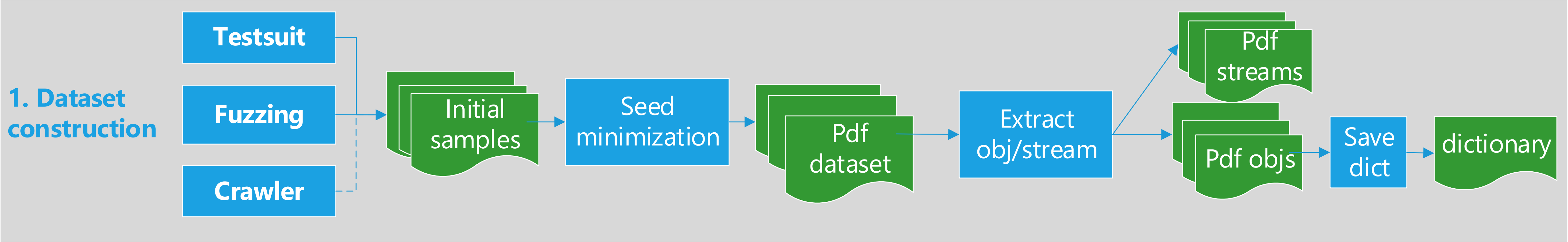
生成

研究方案



1. Dataset construction
Testsuit
Fuzzing
Crawler
Initial samples
Seed minimization
Pdf dataset
Extract obj/stream
Pdf streams
Pdf objs
Save dict
dictionary
2. Training
Cut the text
Vectorization
LSTM, BLSTM, Attention, etc.
Build model
Training
Structure, parameters
Model
3. Generation
Prefix string
Vectorization
Load model
predict
New objs
Attach to pdf host
New pdf files
4. Testing
Target APP
Foxit Reader, Adobe Reader, Mupdf, Chrome, Edge et al.
Fuzzing
Effectiveness measure
Coverage
Bugs

数据集构建（DATASET CONSTRUCTION）



初始PDF样本集：（Testsuite+Fuzzing）

Stillhq.com PDF Database/Mikail's PDF database

QualityLogic’s PDF 1.7 Application Test Suite

Adobe PDF test suites

Ghent Working Group Test Suites

PDF cabinet of horrors

Pdfium_tests

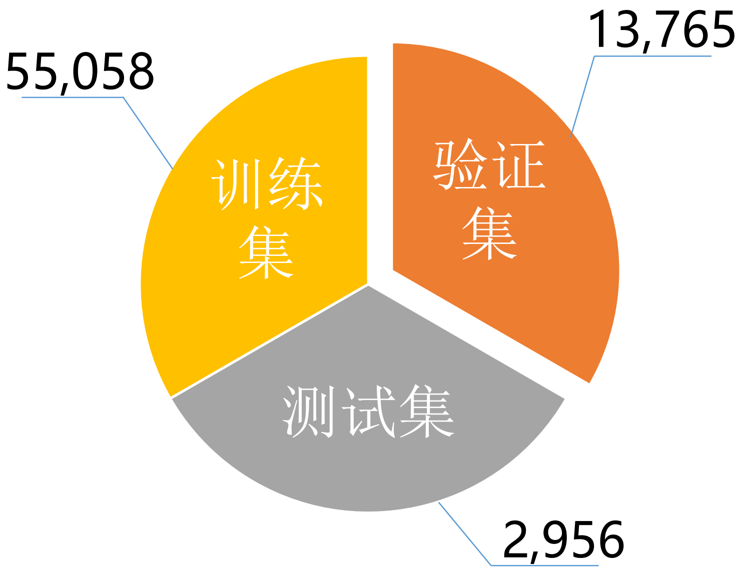
… …

obj总数：**71,779**

stream总数：**23,521**

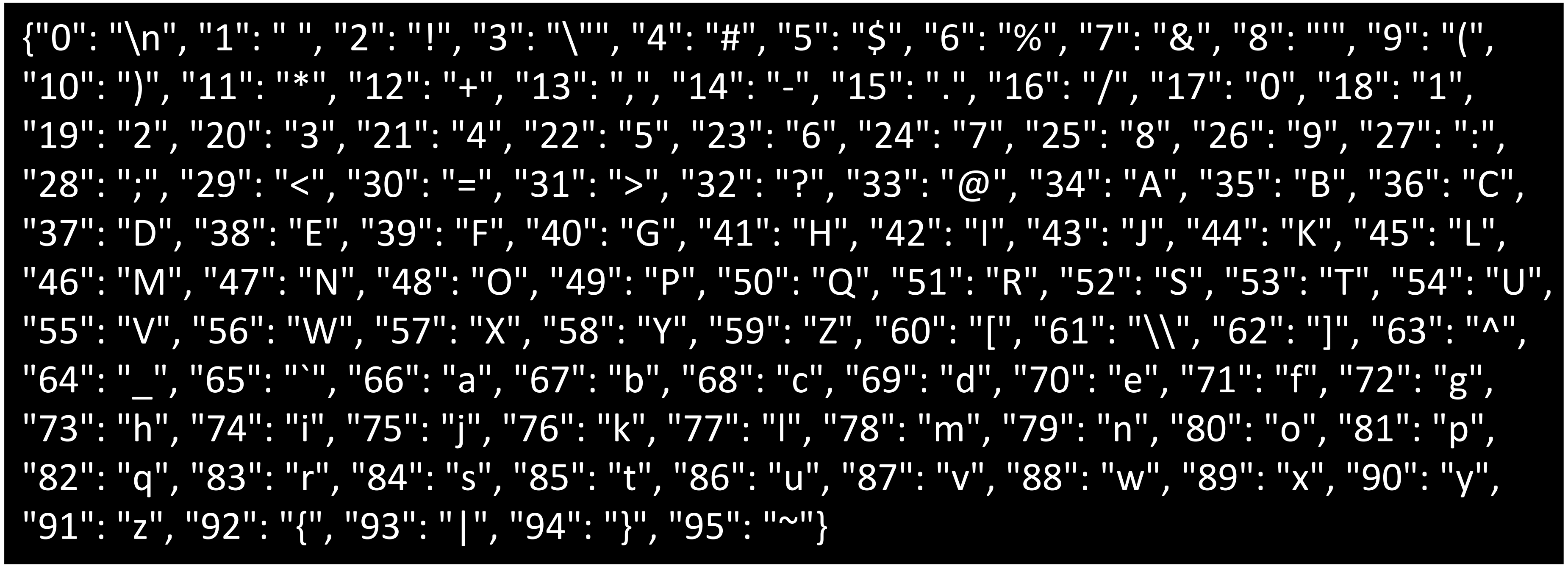
32.77%是带stream的obj

初始样本集	最小集	代码覆盖率
20000+	251	**37.996%**



obj分配

语料字典



```
{"0": "\n", "1": " ", "2": "!", "3": "\"", "4": "#", "5": "$", "6": "%", "7": "&", "8": "'", "9": "(",
"10": ")", "11": "*", "12": "+", "13": ",", "14": "-", "15": ".", "16": "/", "17": "0", "18": "1",
"19": "2", "20": "3", "21": "4", "22": "5", "23": "6", "24": "7", "25": "8", "26": "9", "27": ":",
"28": ";", "29": "<", "30": "=", "31": ">", "32": "?", "33": "@", "34": "A", "35": "B", "36": "C",
"37": "D", "38": "E", "39": "F", "40": "G", "41": "H", "42": "I", "43": "J", "44": "K", "45": "L",
"46": "M", "47": "N", "48": "O", "49": "P", "50": "Q", "51": "R", "52": "S", "53": "T", "54": "U",
"55": "V", "56": "W", "57": "X", "58": "Y", "59": "Z", "60": "[", "61": "\\", "62": "]", "63": "^",
"64": "_", "65": "`", "66": "a", "67": "b", "68": "c", "69": "d", "70": "e", "71": "f", "72": "g",
"73": "h", "74": "i", "75": "j", "76": "k", "77": "l", "78": "m", "79": "n", "80": "o", "81": "p",
"82": "q", "83": "r", "84": "s", "85": "t", "86": "u", "87": "v", "88": "w", "89": "x", "90": "y",
"91": "z", "92": "{", "93": "|", "94": "}", "95": "~"}
```

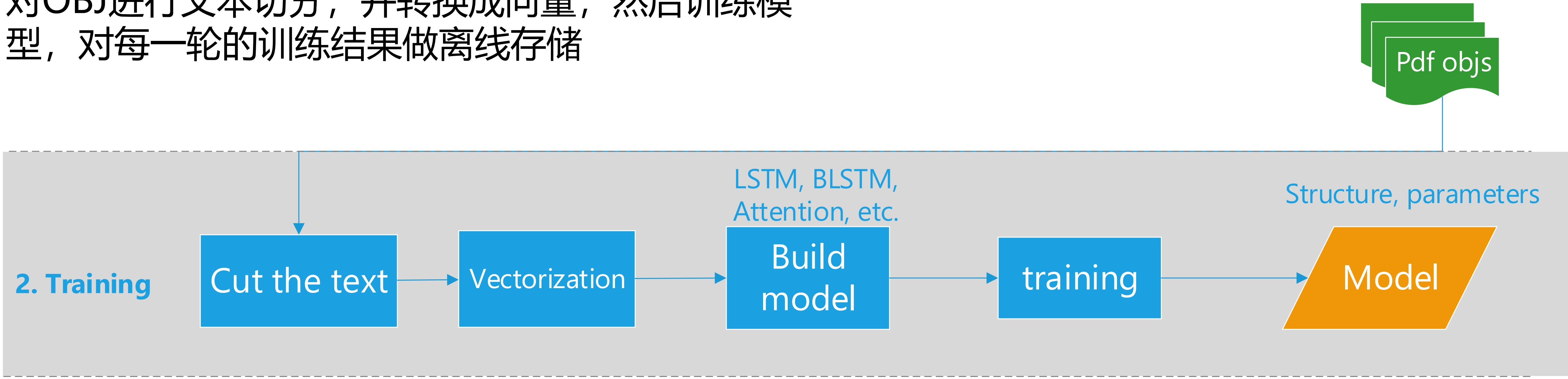
OBJ字典

corpus length：**11,913,817**

total chars: **96**

模型训练（TRAINING）

对OBJ进行文本切分，并转换成向量，然后训练模型，对每一轮的训练结果做离线存储



文本切分（CUT THE TEXT）

总字符数：**11,913,817**

参数设置：maxlen = **50**， step = **3**

切分后总序列数：**3,803,562**(Training:**3,042,849**, validation:**760,713)**

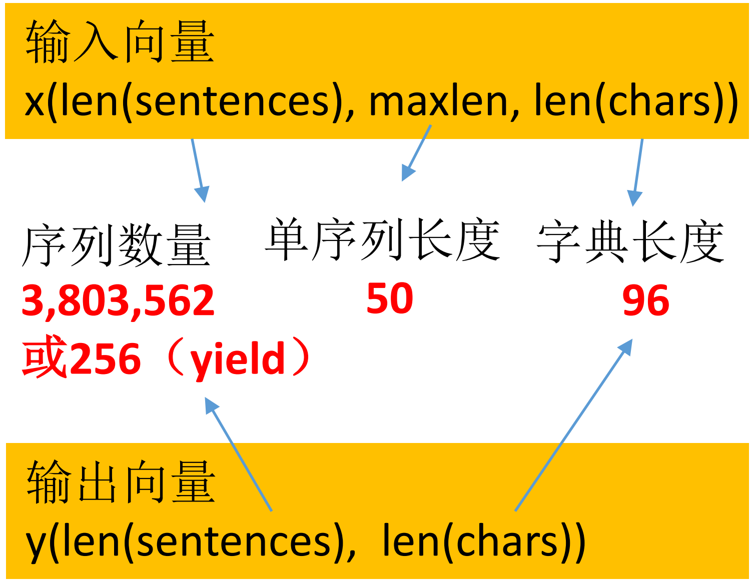
sentences	next_chars
'obj\n<<\n /Type /Page\n /Parent 33 0 R\n /Resources'	' '
'\n<<\n /Type /Page\n /Parent 33 0 R\n /Resources 70'	' '
'\n /Type /Page\n /Parent 33 0 R\n /Resources 70 0 '	'R'
……	……

```
obj
<<
  /Type /Page
  /Parent 33 0 R
  /Resources 70 0 R
  /MediaBox [ 0 0 1247 1984 ]
  /Group <<
    /S /Transparency
    /CS /DeviceRGB
    /I true
  >>
  /Contents 2 0 R
>>
endobj
```

向量化（VECTORIZATION）

编码方式：

One-hot Vector/Encoding



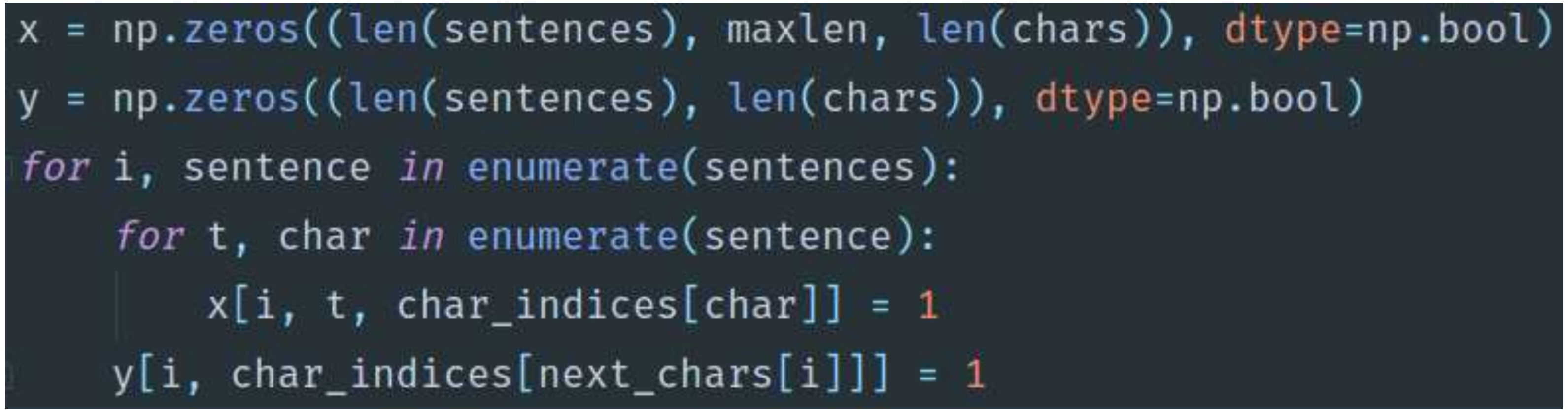
输入向量
x(len(sentences), maxlen, len(chars))

序列数量
3,803,562
或256（yield）

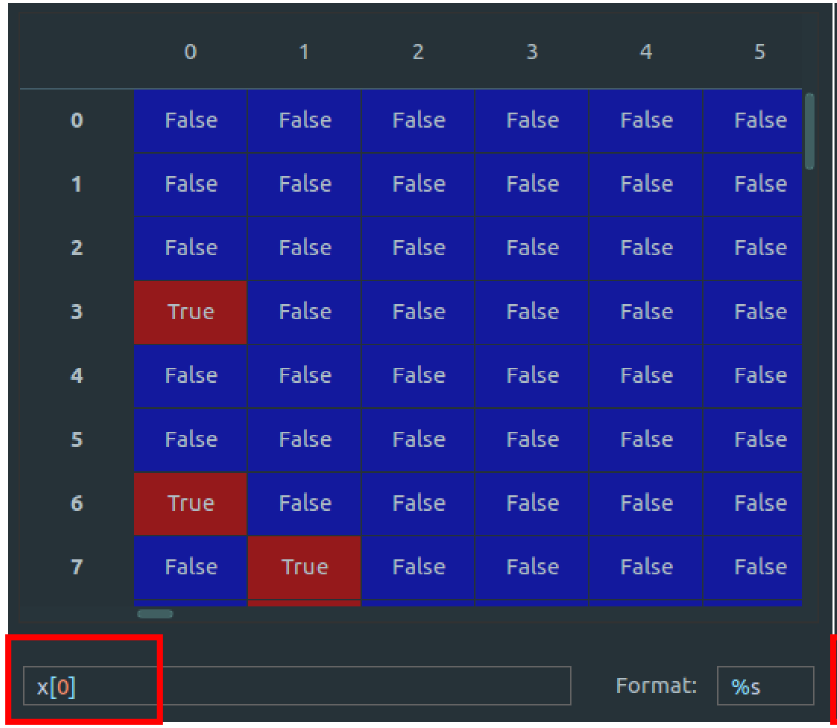
单序列长度
50

字典长度
96

输出向量
y(len(sentences), len(chars))

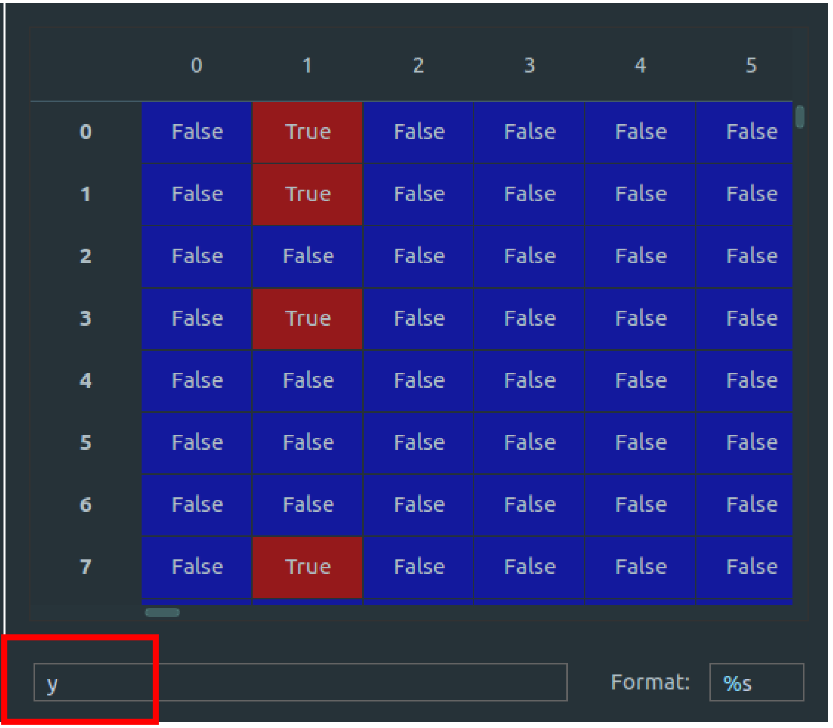


```
x = np.zeros((len(sentences), maxlen, len(chars)), dtype=np.bool)
y = np.zeros((len(sentences), len(chars)), dtype=np.bool)
for i, sentence in enumerate(sentences):
    for t, char in enumerate(sentence):
        x[i, t, char_indices[char]] = 1
    y[i, char_indices[next_chars[i]]] = 1
```



	0	1	2	3	4	5
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	True	False	False	False	False	False
4	False	False	False	False	False	False
5	False	False	False	False	False	False
6	True	False	False	False	False	False
7	False	True	False	False	False	False

x[0] Format: %s

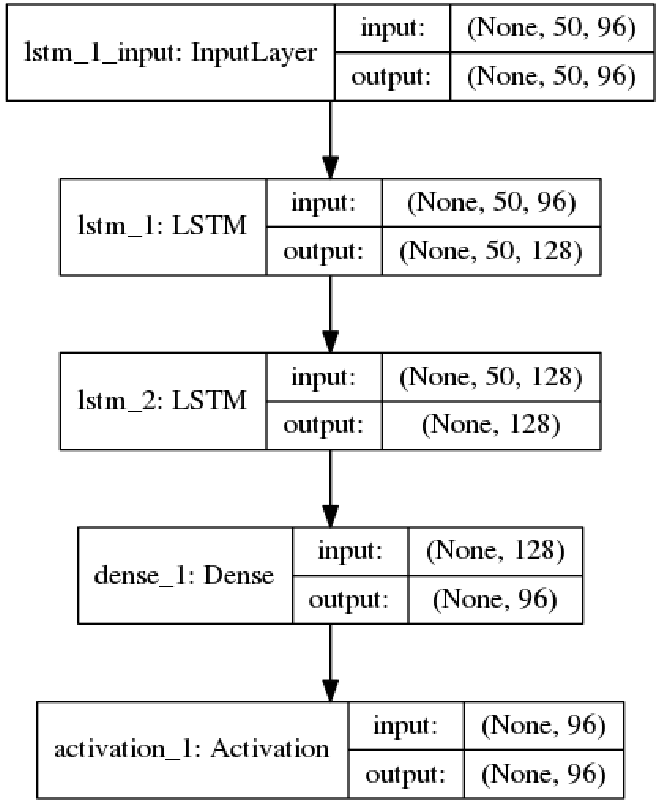


	0	1	2	3	4	5
0	False	True	False	False	False	False
1	False	True	False	False	False	False
2	False	False	False	False	False	False
3	False	True	False	False	False	False
4	False	False	False	False	False	False
5	False	False	False	False	False	False
6	False	False	False	False	False	False
7	False	True	False	False	False	False

y Format: %s

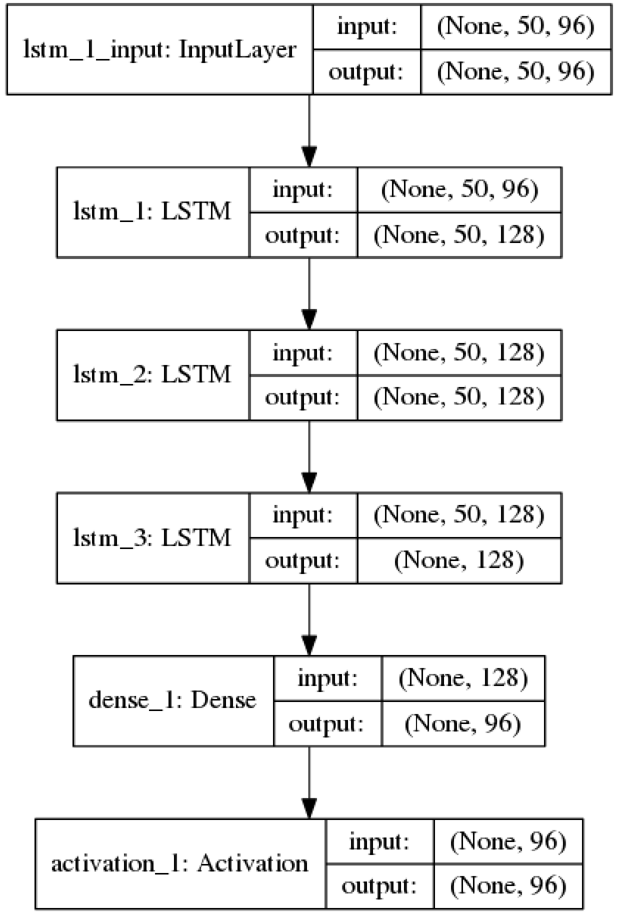
模型设计

```
2LSTM  summary ...
_________________________________________________________________
Layer (type)               Output Shape           Param #
=================================================================
lstm_1 (LSTM)              (None, 50, 128)        115200
_________________________________________________________________
lstm_2 (LSTM)              (None, 128)            131584
_________________________________________________________________
dense_1 (Dense)            (None, 96)             12384
_________________________________________________________________
activation_1 (Activation)  (None, 96)             0
=================================================================
Total params: 259,168
Trainable params: 259,168
Non-trainable params: 0
```



2层LSTM （ **LEARN&FUZZ** 模型）

模型设计



lstm_1_input: InputLayer — input: (None, 50, 96) — output: (None, 50, 96)

lstm_1: LSTM — input: (None, 50, 96) — output: (None, 50, 128)

lstm_2: LSTM — input: (None, 50, 128) — output: (None, 50, 128)

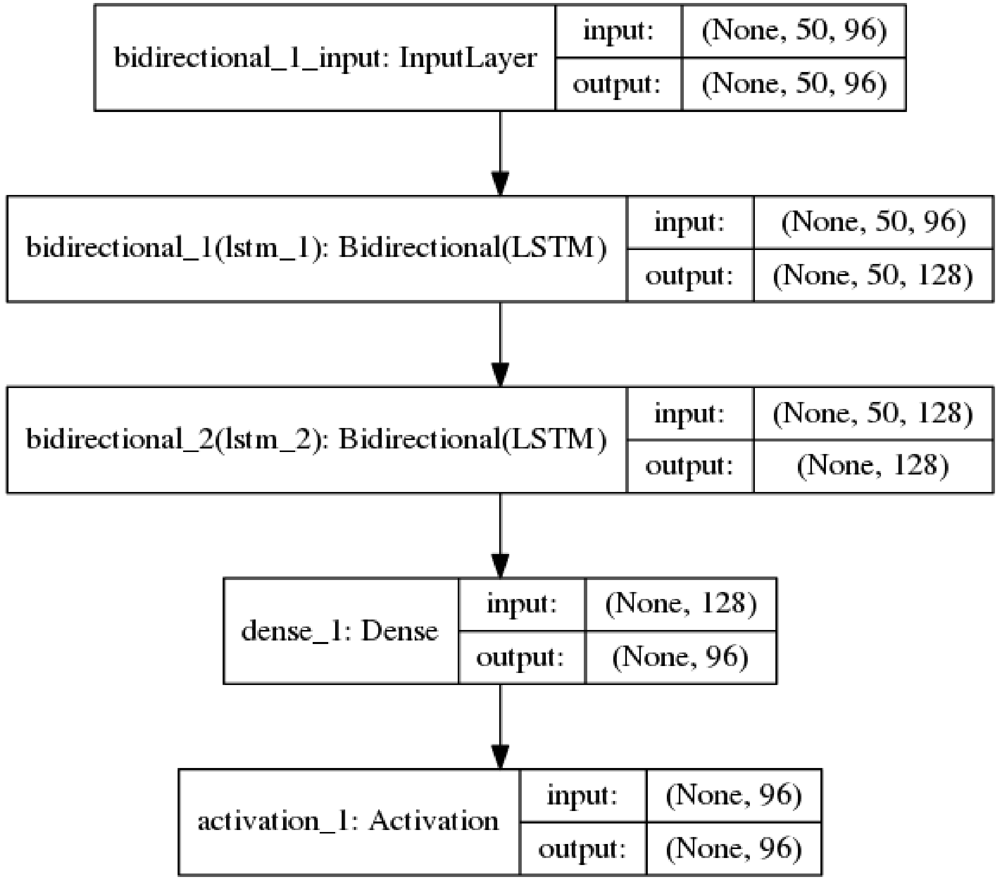
lstm_3: LSTM — input: (None, 50, 128) — output: (None, 128)

dense_1: Dense — input: (None, 128) — output: (None, 96)

activation_1: Activation — input: (None, 96) — output: (None, 96)

3层LSTM

Total params: **390,752**
Trainable params: 390,752
Non-trainable params: 0



bidirectional_1_input: InputLayer — input: (None, 50, 96) — output: (None, 50, 96)

bidirectional_1(lstm_1): Bidirectional(LSTM) — input: (None, 50, 96) — output: (None, 50, 128)

bidirectional_2(lstm_2): Bidirectional(LSTM) — input: (None, 50, 128) — output: (None, 128)

dense_1: Dense — input: (None, 128) — output: (None, 96)

activation_1: Activation — input: (None, 96) — output: (None, 96)

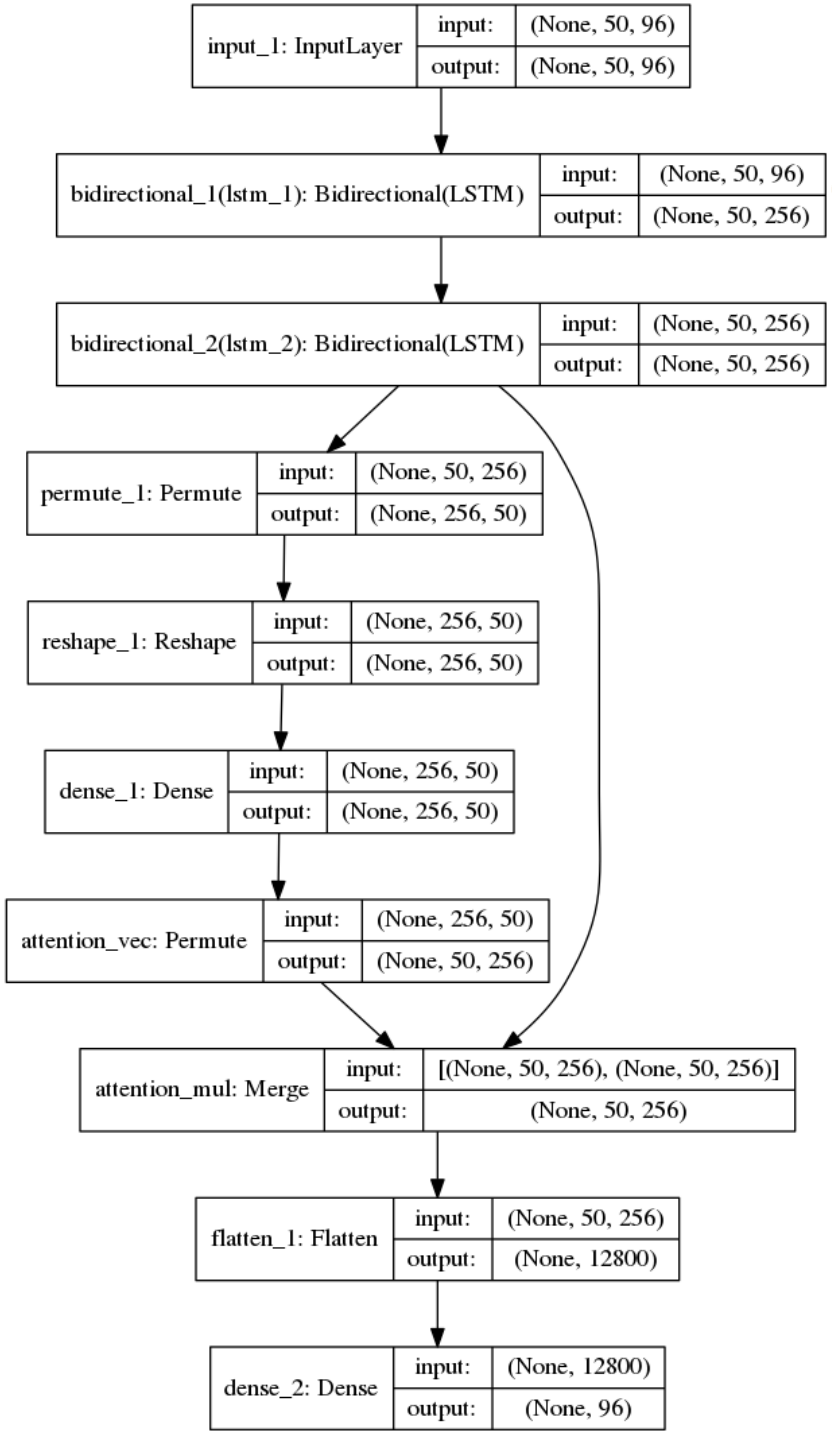
2层BLSTM

Total params: **505,952**
Trainable params: 505,952
Non-trainable params: 0

模型设计

ATTENTION + 2层BLSTM

Total params: **1,856,086**
Trainable params: 1,856,086
Non-trainable params: 0



input_1: InputLayer | input: (None, 50, 96) | output: (None, 50, 96)

bidirectional_1(lstm_1): Bidirectional(LSTM) | input: (None, 50, 96) | output: (None, 50, 256)

bidirectional_2(lstm_2): Bidirectional(LSTM) | input: (None, 50, 256) | output: (None, 50, 256)

permute_1: Permute | input: (None, 50, 256) | output: (None, 256, 50)

reshape_1: Reshape | input: (None, 256, 50) | output: (None, 256, 50)

dense_1: Dense | input: (None, 256, 50) | output: (None, 256, 50)

attention_vec: Permute | input: (None, 256, 50) | output: (None, 50, 256)

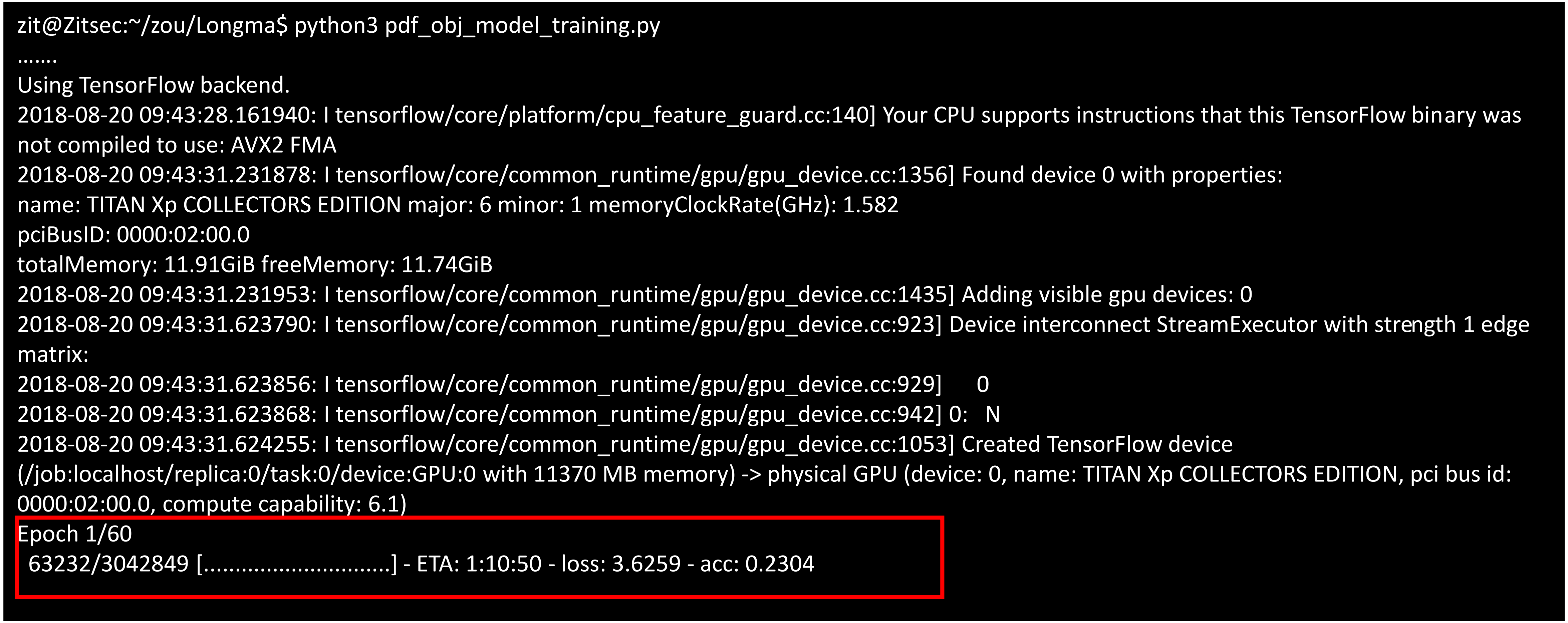
attention_mul: Merge | input: [(None, 50, 256), (None, 50, 256)] | output: (None, 50, 256)

flatten_1: Flatten | input: (None, 50, 256) | output: (None, 12800)

dense_2: Dense | input: (None, 12800) | output: (None, 96)

训练

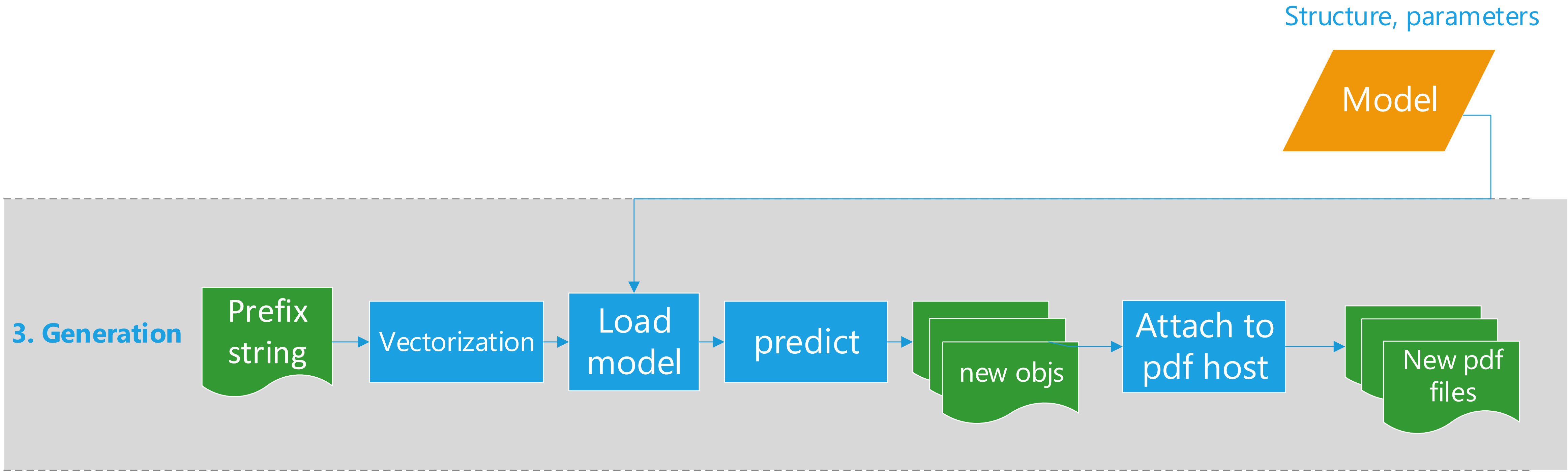
训练参数：batch_size = **256** epoch = **60** optimizer = **adam(lr=1e-4)**，loss=**'categorical_crossentropy'**



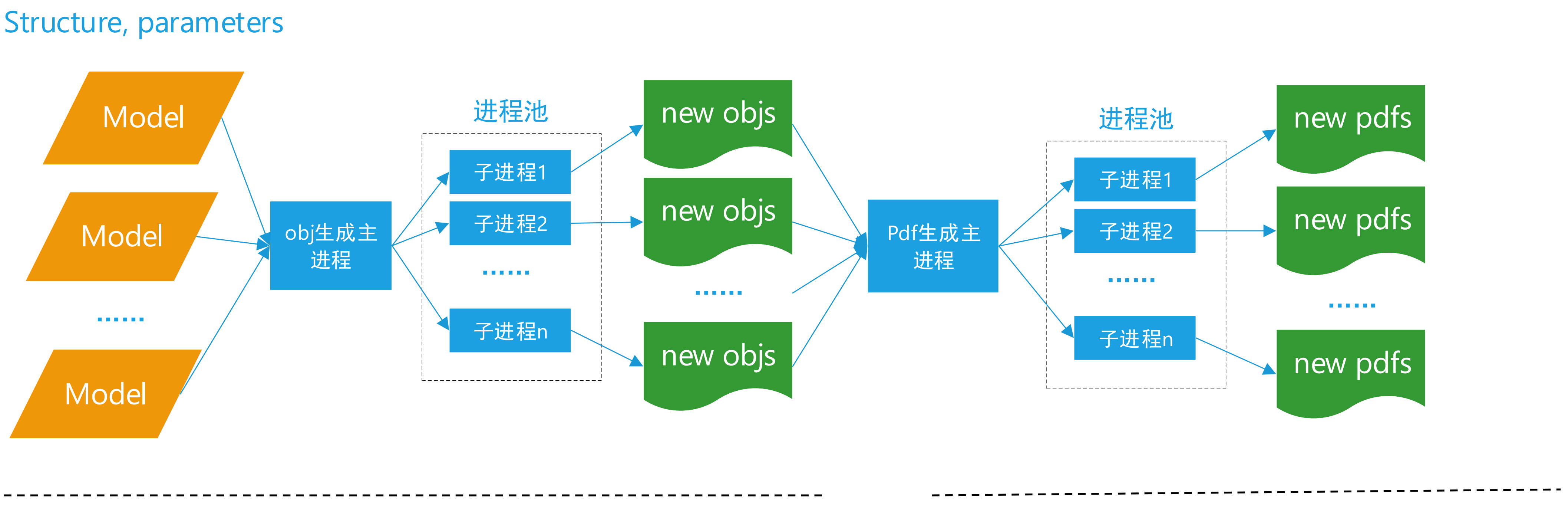
```
zit@Zitsec:~/zou/Longma$ python3 pdf_obj_model_training.py
.......
Using TensorFlow backend.
2018-08-20 09:43:28.161940: I tensorflow/core/platform/cpu_feature_guard.cc:140] Your CPU supports instructions that this TensorFlow binary was
not compiled to use: AVX2 FMA
2018-08-20 09:43:31.231878: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1356] Found device 0 with properties:
name: TITAN Xp COLLECTORS EDITION major: 6 minor: 1 memoryClockRate(GHz): 1.582
pciBusID: 0000:02:00.0
totalMemory: 11.91GiB freeMemory: 11.74GiB
2018-08-20 09:43:31.231953: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1435] Adding visible gpu devices: 0
2018-08-20 09:43:31.623790: I tensorflow/core/common_runtime/gpu/gpu_device.cc:923] Device interconnect StreamExecutor with strength 1 edge
matrix:
2018-08-20 09:43:31.623856: I tensorflow/core/common_runtime/gpu/gpu_device.cc:929]      0
2018-08-20 09:43:31.623868: I tensorflow/core/common_runtime/gpu/gpu_device.cc:942] 0:   N
2018-08-20 09:43:31.624255: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1053] Created TensorFlow device
(/job:localhost/replica:0/task:0/device:GPU:0 with 11370 MB memory) -> physical GPU (device: 0, name: TITAN Xp COLLECTORS EDITION, pci bus id:
0000:02:00.0, compute capability: 6.1)
Epoch 1/60
  63232/3042849 [.............................] - ETA: 1:10:50 - loss: 3.6259 - acc: 0.2304
```

生成（GENERATION）

选取PREFIX STRING，向量化，加载模型，预测生成OBJ，并由OBJ生成PDF



生成（GENERATION）



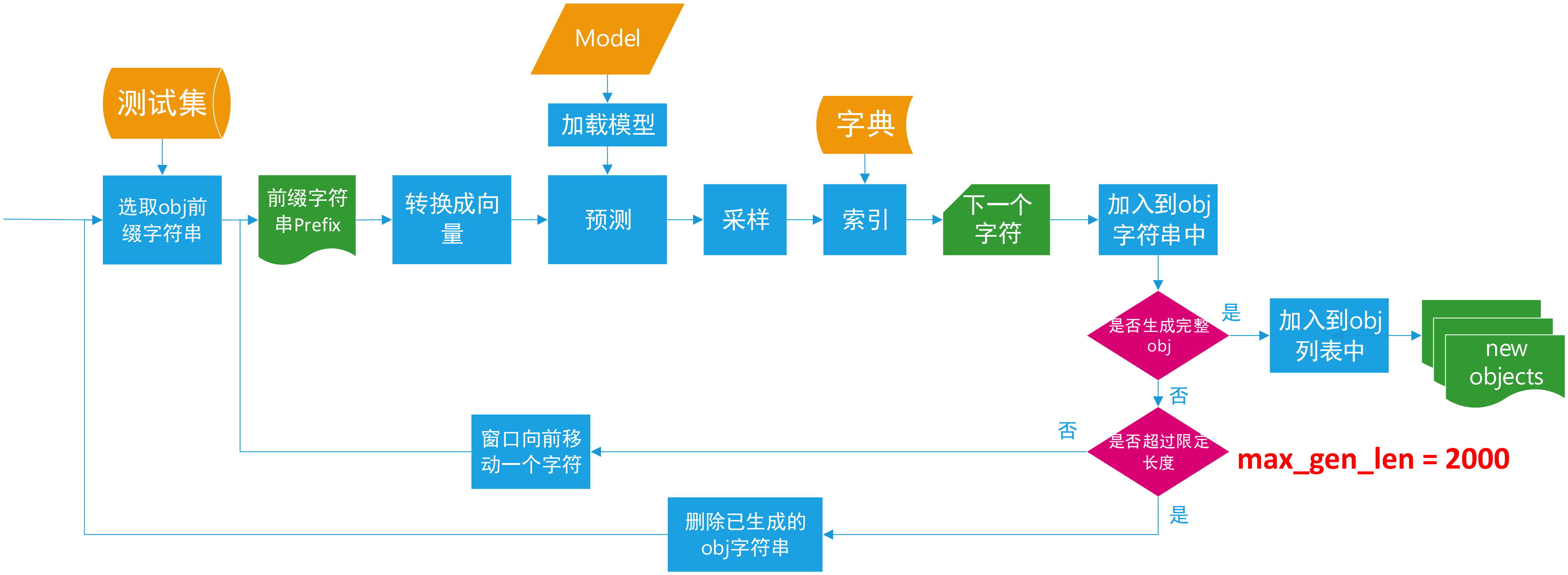
OBJ生成

PDF生成

样本生成阶段的两个重要的**进程池**

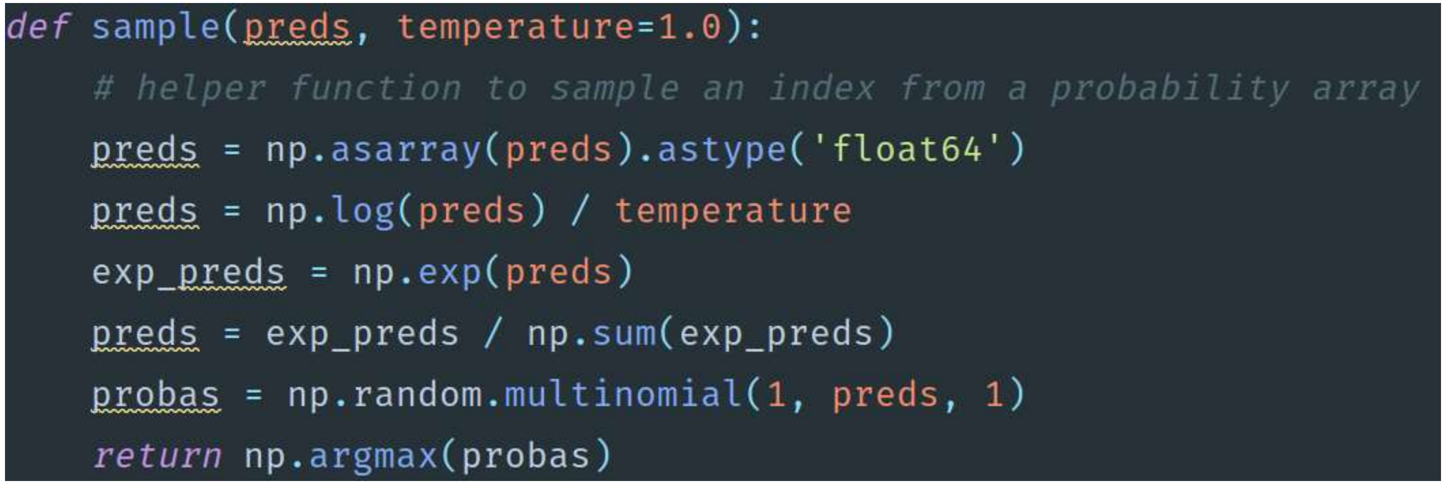
- 并行加载多个模型进行推断
- 并行生成多批次的obj和PDF样本
- 缩短实验周期，增强模型的可扩展性

OBJ生成



- 若生成完整OBJ，则加入列表中；
- 若生成长度超过阈值，则回退、丢弃已生成的字符，重新从测试集中选择PREFIX生成

OBJ生成



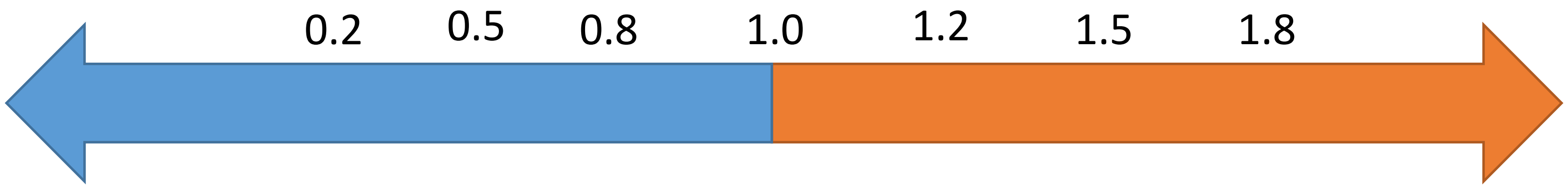
```
def sample(preds, temperature=1.0):
    # helper function to sample an index from a probability array
    preds = np.asarray(preds).astype('float64')
    preds = np.log(preds) / temperature
    exp_preds = np.exp(preds)
    preds = exp_preds / np.sum(exp_preds)
    probas = np.random.multinomial(1, preds, 1)
    return np.argmax(probas)
```

采样函数

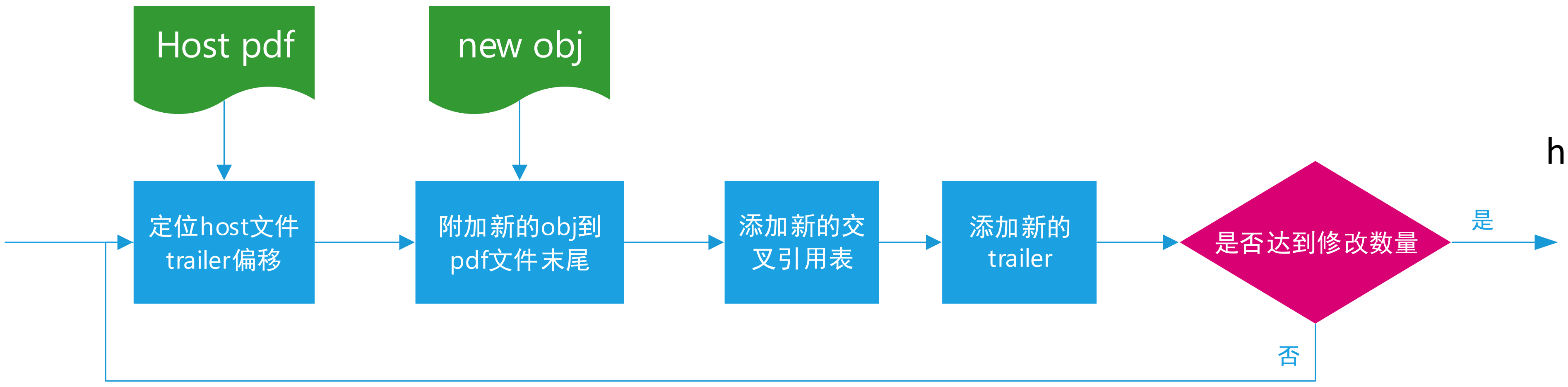
概率分布差异性变大，生成文本有序性变强，更接近真实值的数据

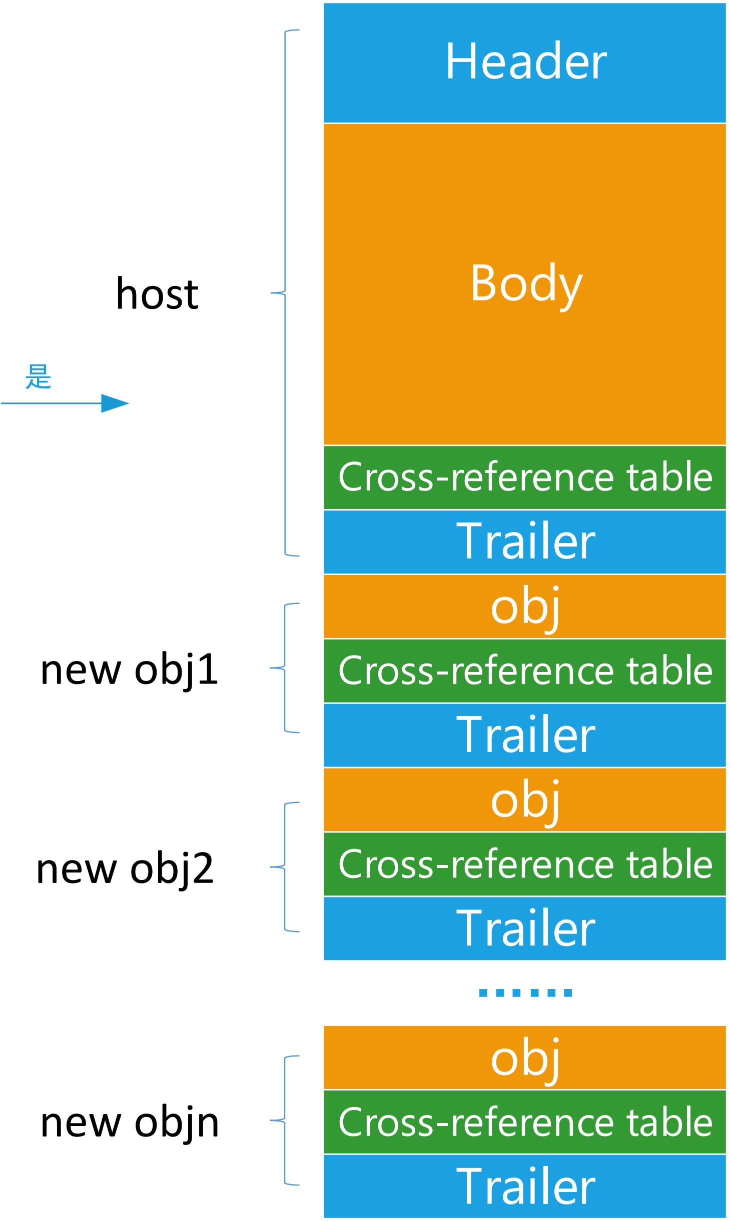
temperature

概率分布差异性变小，生成文本随机性变强，趋向于多样性、随机的数据



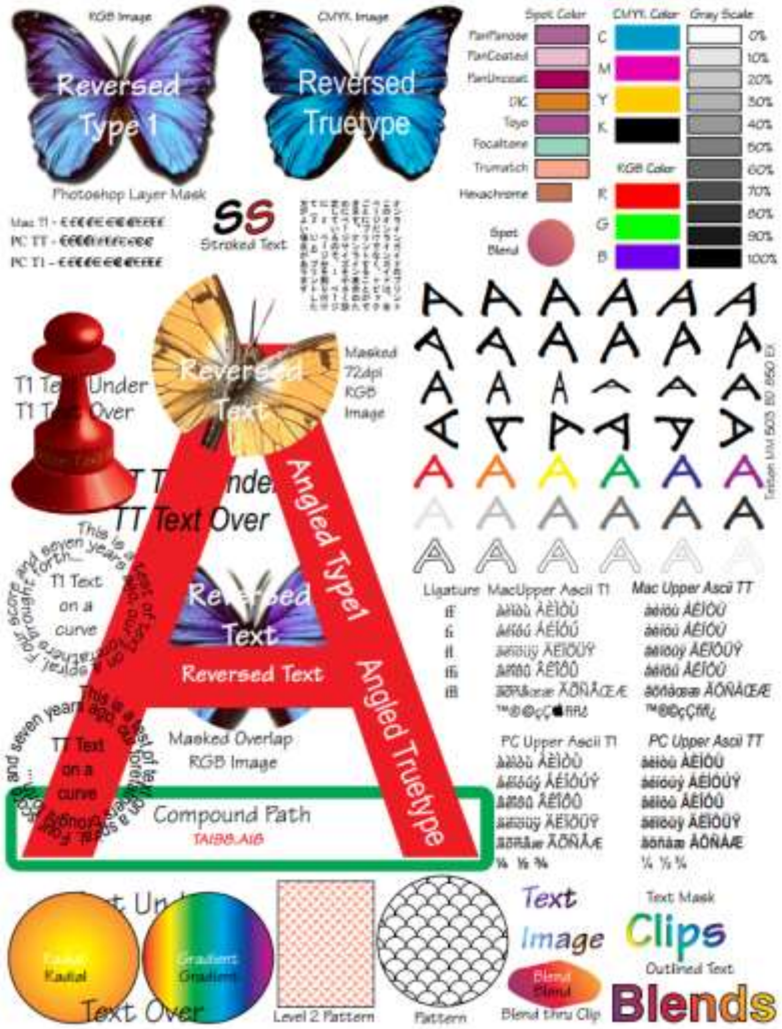
PDF生成



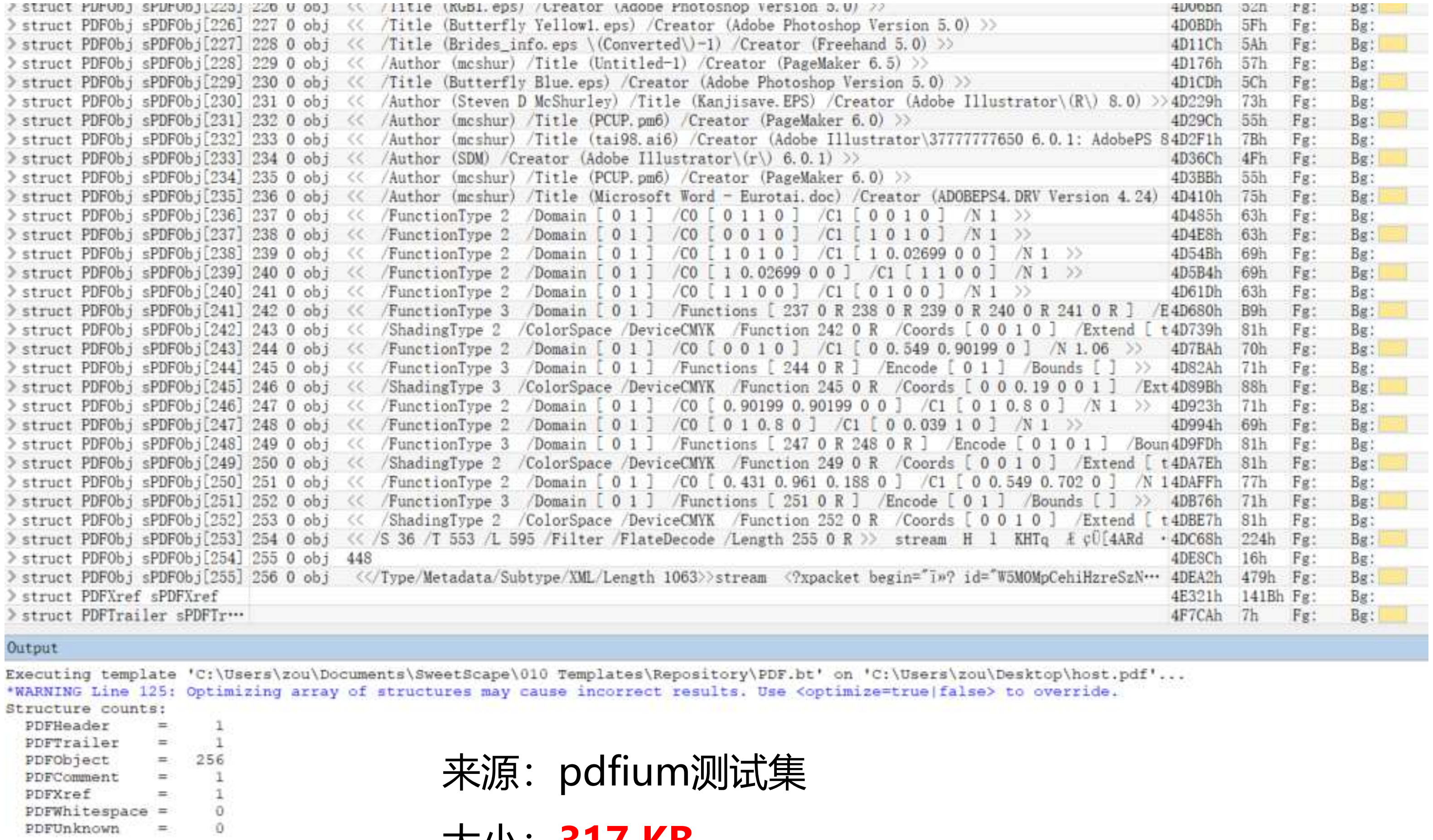


以**增量更新**（Incremental update）的方式把新生成的obj附加到host文件的末尾，实现对host文件中obj的更新和替换

PDF生成



宿主文件（HOST）



```
> struct PDFObj sPDFObj[225] 226 0 obj  << /Title (RGB1.eps) /Creator (Adobe Photoshop Version 5.0) >>                                4D06Bh  52h   Fg:  Bg:
> struct PDFObj sPDFObj[226] 227 0 obj  << /Title (Butterfly Yellow1.eps) /Creator (Adobe Photoshop Version 5.0) >>                   4D0BDh  5Fh   Fg:  Bg:
> struct PDFObj sPDFObj[227] 228 0 obj  << /Title (Brides_info.eps \(Converted\)-1) /Creator (Freehand 5.0) >>                        4D11Ch  5Ah   Fg:  Bg:
> struct PDFObj sPDFObj[228] 229 0 obj  << /Author (mcshur) /Title (Untitled-1) /Creator (PageMaker 6.5) >>                           4D176h  57h   Fg:  Bg:
> struct PDFObj sPDFObj[229] 230 0 obj  << /Title (Butterfly Blue.eps) /Creator (Adobe Photoshop Version 5.0) >>                      4D1CDh  5Ch   Fg:  Bg:
> struct PDFObj sPDFObj[230] 231 0 obj  << /Author (Steven D McShurley) /Title (Kanjisave.EPS) /Creator (Adobe Illustrator\(R\) 8.0) >>4D229h  73h   Fg:  Bg:
> struct PDFObj sPDFObj[231] 232 0 obj  << /Author (mcshur) /Title (PCUP.pm6) /Creator (PageMaker 6.0) >>                             4D29Ch  55h   Fg:  Bg:
> struct PDFObj sPDFObj[232] 233 0 obj  << /Author (mcshur) /Title (tai98.ai6) /Creator (Adobe Illustrator\37777777650 6.0.1: AdobePS 84D2F1h  7Bh   Fg:  Bg:
> struct PDFObj sPDFObj[233] 234 0 obj  << /Author (SDM) /Creator (Adobe Illustrator\(r\) 6.0.1) >>                                   4D36Ch  4Fh   Fg:  Bg:
> struct PDFObj sPDFObj[234] 235 0 obj  << /Author (mcshur) /Title (PCUP.pm6) /Creator (PageMaker 6.0) >>                             4D3BBh  55h   Fg:  Bg:
> struct PDFObj sPDFObj[235] 236 0 obj  << /Author (mcshur) /Title (Microsoft Word - Eurotai.doc) /Creator (ADOBEPS4.DRV Version 4.24) 4D410h  75h   Fg:  Bg:
> struct PDFObj sPDFObj[236] 237 0 obj  << /FunctionType 2  /Domain [ 0 1 ]  /C0 [ 0 1 1 0 ]  /C1 [ 0 0 1 0 ]  /N 1  >>               4D485h  63h   Fg:  Bg:
> struct PDFObj sPDFObj[237] 238 0 obj  << /FunctionType 2  /Domain [ 0 1 ]  /C0 [ 0 0 1 0 ]  /C1 [ 1 0 1 0 ]  /N 1  >>               4D4E8h  63h   Fg:  Bg:
> struct PDFObj sPDFObj[238] 239 0 obj  << /FunctionType 2  /Domain [ 0 1 ]  /C0 [ 1 0 1 0 ]  /C1 [ 1 0.02699 0 0 ]  /N 1  >>         4D54Bh  69h   Fg:  Bg:
> struct PDFObj sPDFObj[239] 240 0 obj  << /FunctionType 2  /Domain [ 0 1 ]  /C0 [ 1 0.02699 0 0 ]  /C1 [ 1 1 0 0 ]  /N 1  >>         4D5B4h  69h   Fg:  Bg:
> struct PDFObj sPDFObj[240] 241 0 obj  << /FunctionType 2  /Domain [ 0 1 ]  /C0 [ 1 1 0 0 ]  /C1 [ 0 1 0 0 ]  /N 1  >>               4D61Dh  63h   Fg:  Bg:
> struct PDFObj sPDFObj[241] 242 0 obj  << /FunctionType 3  /Domain [ 0 1 ]  /Functions [ 237 0 R 238 0 R 239 0 R 240 0 R 241 0 R ]  /E4D680h  B9h   Fg:  Bg:
> struct PDFObj sPDFObj[242] 243 0 obj  << /ShadingType 2  /ColorSpace /DeviceCMYK  /Function 242 0 R  /Coords [ 0 0 1 0 ]  /Extend [ t4D739h  81h   Fg:  Bg:
> struct PDFObj sPDFObj[243] 244 0 obj  << /FunctionType 2  /Domain [ 0 1 ]  /C0 [ 0 0 1 0 ]  /C1 [ 0 0.549 0.90199 0 ]  /N 1.06  >>   4D7BAh  70h   Fg:  Bg:
> struct PDFObj sPDFObj[244] 245 0 obj  << /FunctionType 3  /Domain [ 0 1 ]  /Functions [ 244 0 R ]  /Encode [ 0 1 ]  /Bounds [ ]  >>  4D82Ah  71h   Fg:  Bg:
> struct PDFObj sPDFObj[245] 246 0 obj  << /ShadingType 3  /ColorSpace /DeviceCMYK  /Function 245 0 R  /Coords [ 0 0 0.19 0 0 1 ]  /Ext4D89Bh  88h   Fg:  Bg:
> struct PDFObj sPDFObj[246] 247 0 obj  << /FunctionType 2  /Domain [ 0 1 ]  /C0 [ 0.90199 0.90199 0 0 ]  /C1 [ 0 1 0.8 0 ]  /N 1  >>  4D923h  71h   Fg:  Bg:
> struct PDFObj sPDFObj[247] 248 0 obj  << /FunctionType 2  /Domain [ 0 1 ]  /C0 [ 0 1 0.8 0 ]  /C1 [ 0 0.039 1 0 ]  /N 1  >>          4D994h  69h   Fg:  Bg:
> struct PDFObj sPDFObj[248] 249 0 obj  << /FunctionType 3  /Domain [ 0 1 ]  /Functions [ 247 0 R 248 0 R ]  /Encode [ 0 1 0 1 ]  /Boun4D9FDh  81h   Fg:  Bg:
> struct PDFObj sPDFObj[249] 250 0 obj  << /ShadingType 2  /ColorSpace /DeviceCMYK  /Function 249 0 R  /Coords [ 0 0 1 0 ]  /Extend [ t4DA7Eh  81h   Fg:  Bg:
> struct PDFObj sPDFObj[250] 251 0 obj  << /FunctionType 2  /Domain [ 0 1 ]  /C0 [ 0.431 0.961 0.188 0 ]  /C1 [ 0 0.549 0.702 0 ]  /N 14DAFFh  77h   Fg:  Bg:
> struct PDFObj sPDFObj[251] 252 0 obj  << /FunctionType 3  /Domain [ 0 1 ]  /Functions [ 251 0 R ]  /Encode [ 0 1 ]  /Bounds [ ]  >>  4DB76h  71h   Fg:  Bg:
> struct PDFObj sPDFObj[252] 253 0 obj  << /ShadingType 2  /ColorSpace /DeviceCMYK  /Function 252 0 R  /Coords [ 0 0 1 0 ]  /Extend [ t4DBE7h  81h   Fg:  Bg:
> struct PDFObj sPDFObj[253] 254 0 obj  << /S 36 /T 553 /L 595 /Filter /FlateDecode /Length 255 0 R >>  stream  H  1  KHTq  ‡ çÜ[4ARd  ·4DC68h  224h  Fg:  Bg:
> struct PDFObj sPDFObj[254] 255 0 obj  448                                                                                            4DE8Ch  16h   Fg:  Bg:
> struct PDFObj sPDFObj[255] 256 0 obj   <</Type/Metadata/Subtype/XML/Length 1063>>stream  <?xpacket begin="ï»? id="W5M0MpCehiHzreSzN···  4DEA2h  479h  Fg:  Bg:
> struct PDFXref sPDFXref                                                                                                            4E321h  141Bh Fg:  Bg:
> struct PDFTrailer sPDFTr···                                                                                                        4F7CAh  7h    Fg:  Bg:
```

Output

```
Executing template 'C:\Users\zou\Documents\SweetScape\010 Templates\Repository\PDF.bt' on 'C:\Users\zou\Desktop\host.pdf'...
*WARNING Line 125: Optimizing array of structures may cause incorrect results. Use <optimize=true|false> to override.
Structure counts:
   PDFHeader      =    1
   PDFTrailer     =    1
   PDFObject      =  256
   PDFComment     =    1
   PDFXref        =    1
   PDFWhitespace  =    0
   PDFUnknown     =    0
```

来源：pdfium测试集

大小：**317 KB**

obj总数：**257**

obj替换比例：**1/10**

实验分析

模型训练及样本生成

PDF样本测试

模型训练及样本生成

实验环境

模型训练结果分析

OBJ样本生成结果分析

PDF样本生成结果分析

实验环境

硬件环境

TITAN Xp COLLECTORS EDITION **X4**

E5-2683 v4 **X2**

256G

开发环境

Ubuntu-16.04.2-desktop-amd64

Python 3.5

Keras

前端

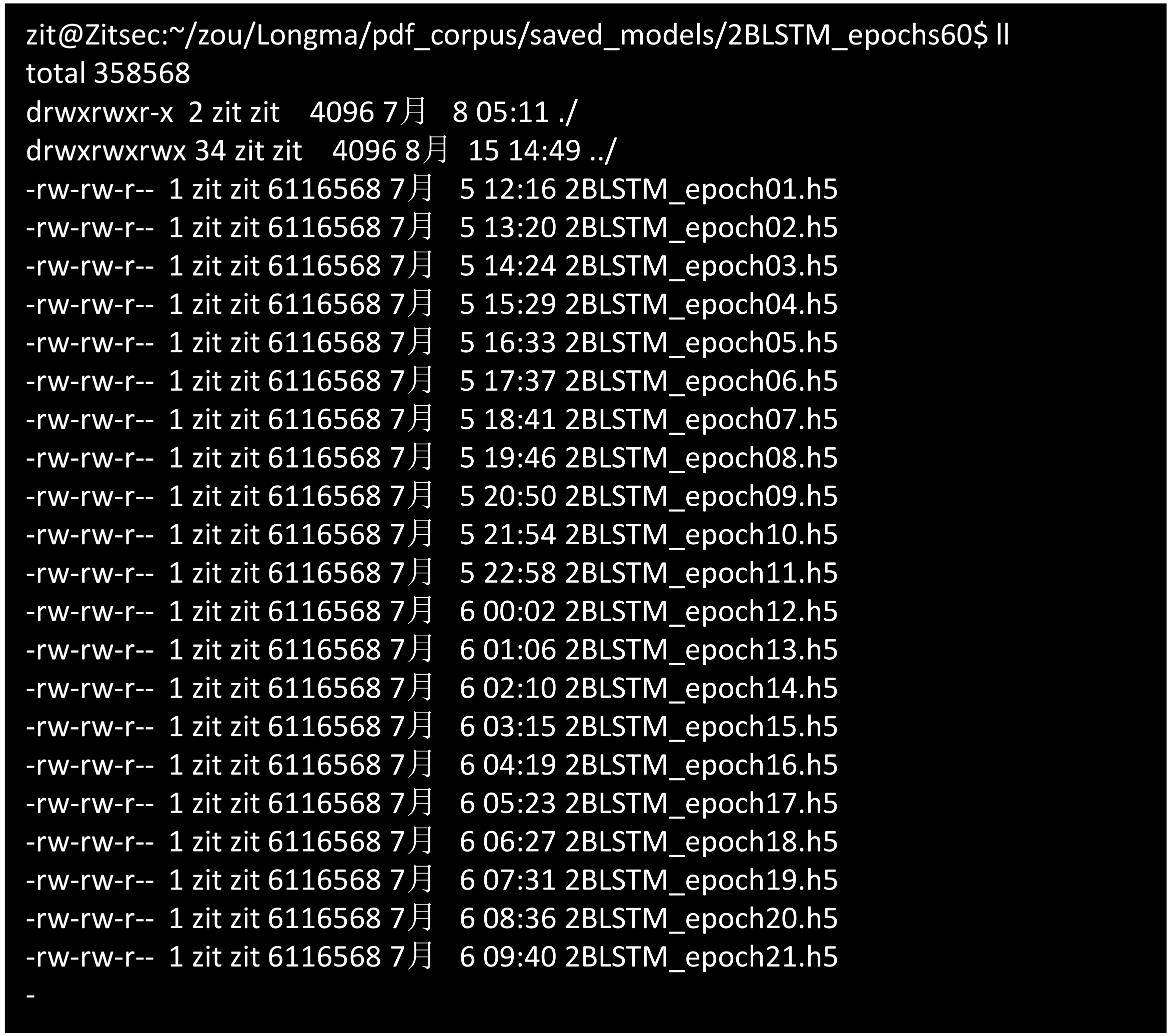
TensorFlow

后端

模型训练结果分析

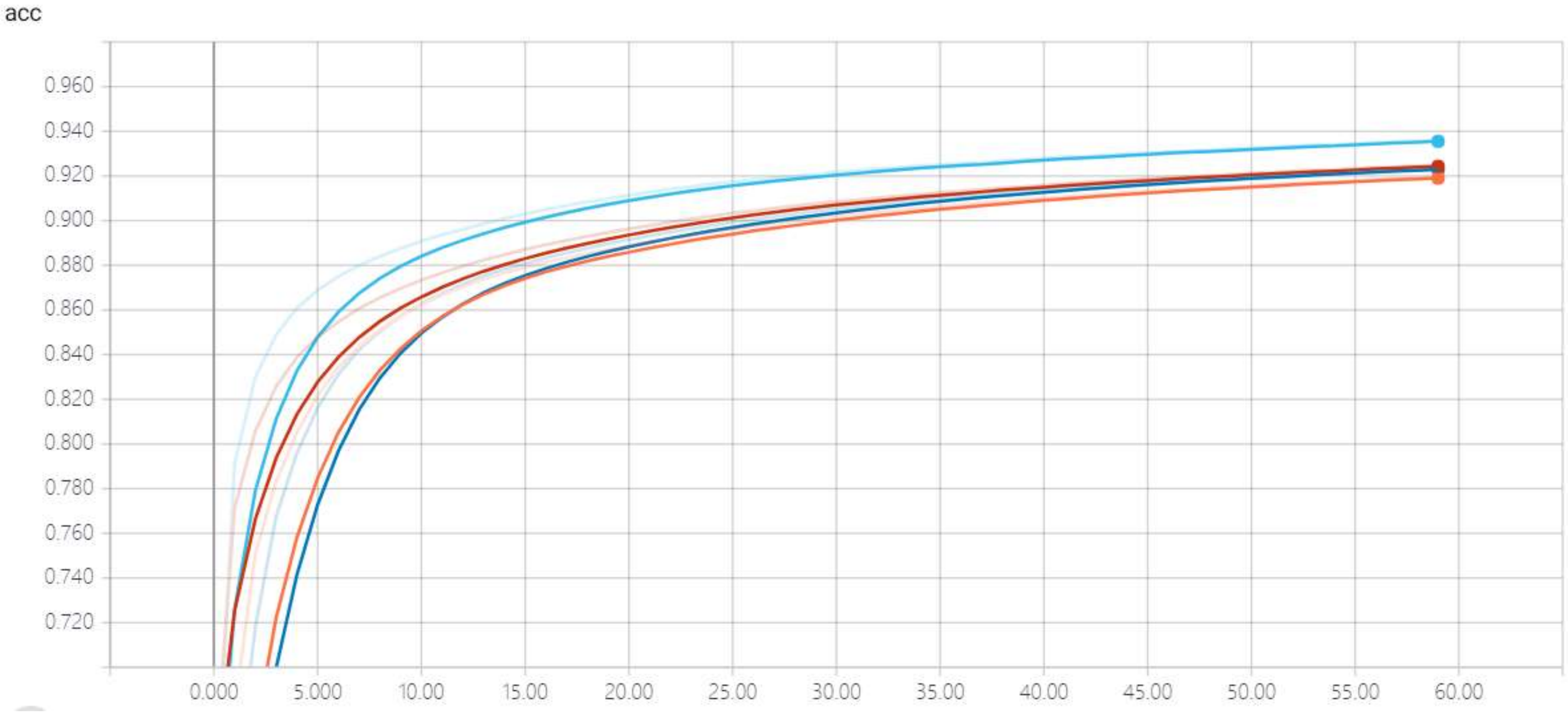
训练轮次：**60**

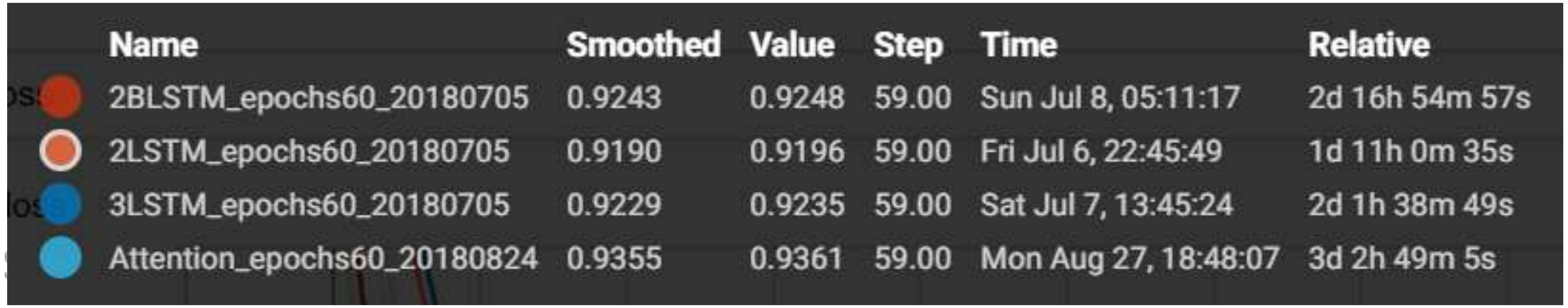
模型	参数	训练时间	模型文件大小（M）
2LSTM	259,168	1d 11h 0m 35s	3.00
3LSTM	390,752	2d 1h 38m 49s	4.51
2BLSTM	505,952	2d 16h 54m 57s	5.83
Attention	1,800,786	3d 2h 49m 5s	21.30



```
zit@Zitsec:~/zou/Longma/pdf_corpus/saved_models/2BLSTM_epochs60$ ll
total 358568
drwxrwxr-x  2 zit zit    4096 7月   8 05:11 ./
drwxrwxrwx 34 zit zit    4096 8月  15 14:49 ../
-rw-rw-r--  1 zit zit 6116568 7月   5 12:16 2BLSTM_epoch01.h5
-rw-rw-r--  1 zit zit 6116568 7月   5 13:20 2BLSTM_epoch02.h5
-rw-rw-r--  1 zit zit 6116568 7月   5 14:24 2BLSTM_epoch03.h5
-rw-rw-r--  1 zit zit 6116568 7月   5 15:29 2BLSTM_epoch04.h5
-rw-rw-r--  1 zit zit 6116568 7月   5 16:33 2BLSTM_epoch05.h5
-rw-rw-r--  1 zit zit 6116568 7月   5 17:37 2BLSTM_epoch06.h5
-rw-rw-r--  1 zit zit 6116568 7月   5 18:41 2BLSTM_epoch07.h5
-rw-rw-r--  1 zit zit 6116568 7月   5 19:46 2BLSTM_epoch08.h5
-rw-rw-r--  1 zit zit 6116568 7月   5 20:50 2BLSTM_epoch09.h5
-rw-rw-r--  1 zit zit 6116568 7月   5 21:54 2BLSTM_epoch10.h5
-rw-rw-r--  1 zit zit 6116568 7月   5 22:58 2BLSTM_epoch11.h5
-rw-rw-r--  1 zit zit 6116568 7月   6 00:02 2BLSTM_epoch12.h5
-rw-rw-r--  1 zit zit 6116568 7月   6 01:06 2BLSTM_epoch13.h5
-rw-rw-r--  1 zit zit 6116568 7月   6 02:10 2BLSTM_epoch14.h5
-rw-rw-r--  1 zit zit 6116568 7月   6 03:15 2BLSTM_epoch15.h5
-rw-rw-r--  1 zit zit 6116568 7月   6 04:19 2BLSTM_epoch16.h5
-rw-rw-r--  1 zit zit 6116568 7月   6 05:23 2BLSTM_epoch17.h5
-rw-rw-r--  1 zit zit 6116568 7月   6 06:27 2BLSTM_epoch18.h5
-rw-rw-r--  1 zit zit 6116568 7月   6 07:31 2BLSTM_epoch19.h5
-rw-rw-r--  1 zit zit 6116568 7月   6 08:36 2BLSTM_epoch20.h5
-rw-rw-r--  1 zit zit 6116568 7月   6 09:40 2BLSTM_epoch21.h5
-
```

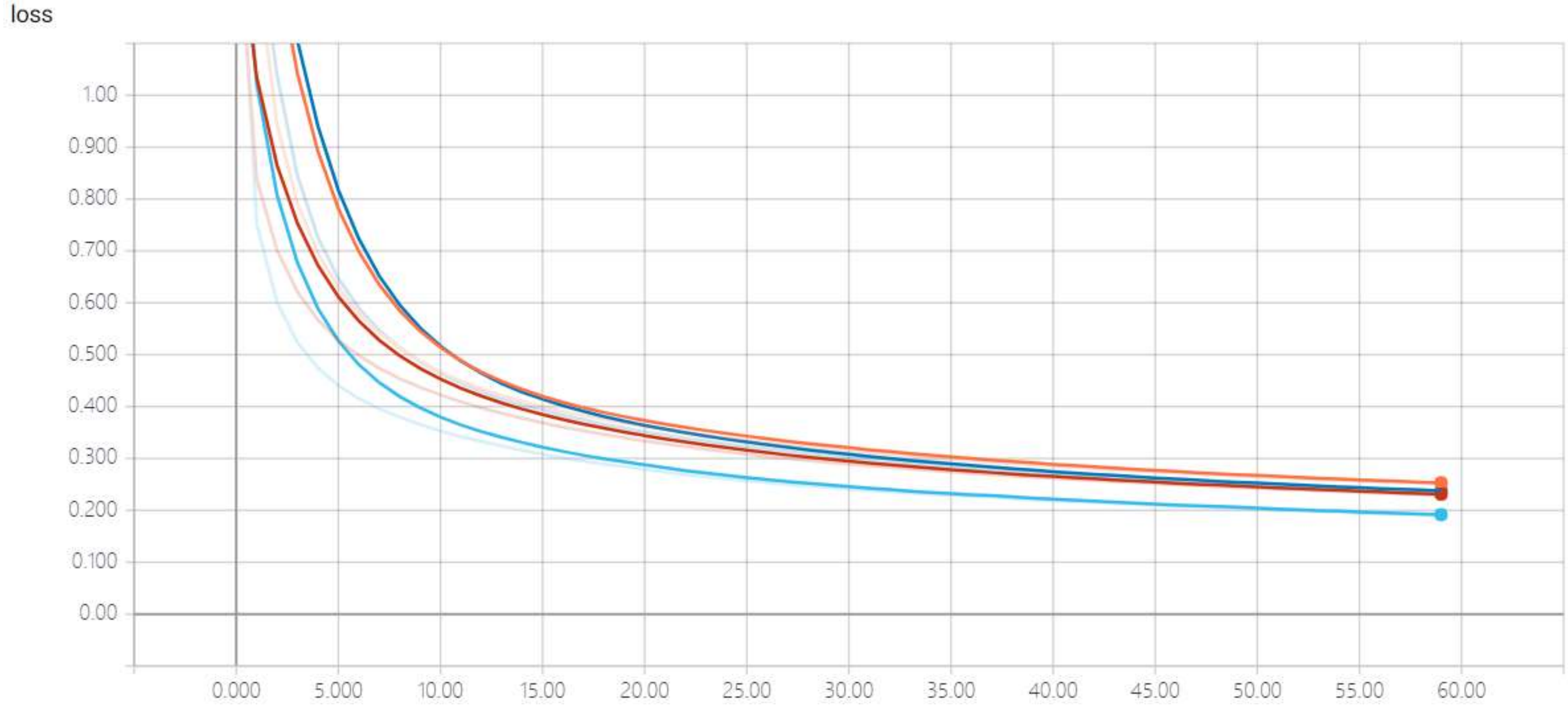
ACC曲线

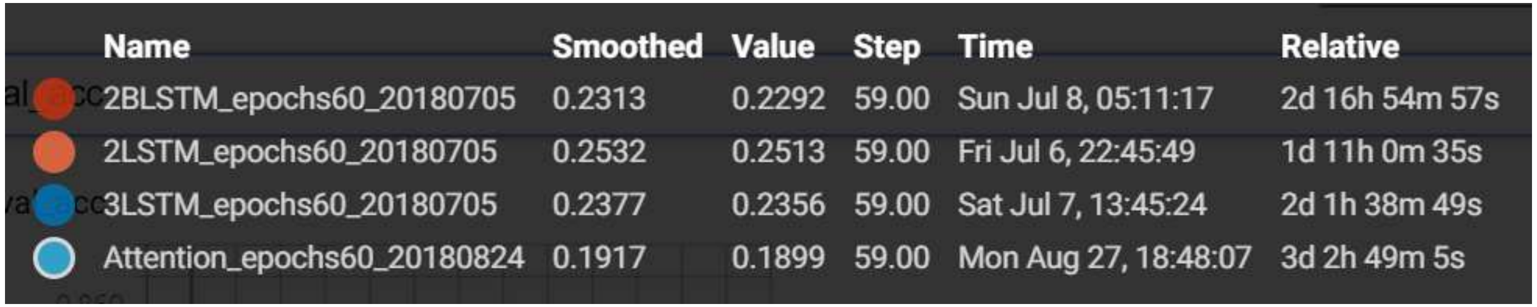




	Name	Smoothed	Value	Step	Time	Relative
●	2BLSTM_epochs60_20180705	0.9243	0.9248	59.00	Sun Jul 8, 05:11:17	2d 16h 54m 57s
●	2LSTM_epochs60_20180705	0.9190	0.9196	59.00	Fri Jul 6, 22:45:49	1d 11h 0m 35s
●	3LSTM_epochs60_20180705	0.9229	0.9235	59.00	Sat Jul 7, 13:45:24	2d 1h 38m 49s
●	Attention_epochs60_20180824	0.9355	0.9361	59.00	Mon Aug 27, 18:48:07	3d 2h 49m 5s

LOSS曲线





	Name	Smoothed	Value	Step	Time	Relative
●	2BLSTM_epochs60_20180705	0.2313	0.2292	59.00	Sun Jul 8, 05:11:17	2d 16h 54m 57s
●	2LSTM_epochs60_20180705	0.2532	0.2513	59.00	Fri Jul 6, 22:45:49	1d 11h 0m 35s
●	3LSTM_epochs60_20180705	0.2377	0.2356	59.00	Sat Jul 7, 13:45:24	2d 1h 38m 49s
○	Attention_epochs60_20180824	0.1917	0.1899	59.00	Mon Aug 27, 18:48:07	3d 2h 49m 5s

OBJ生成结果分析

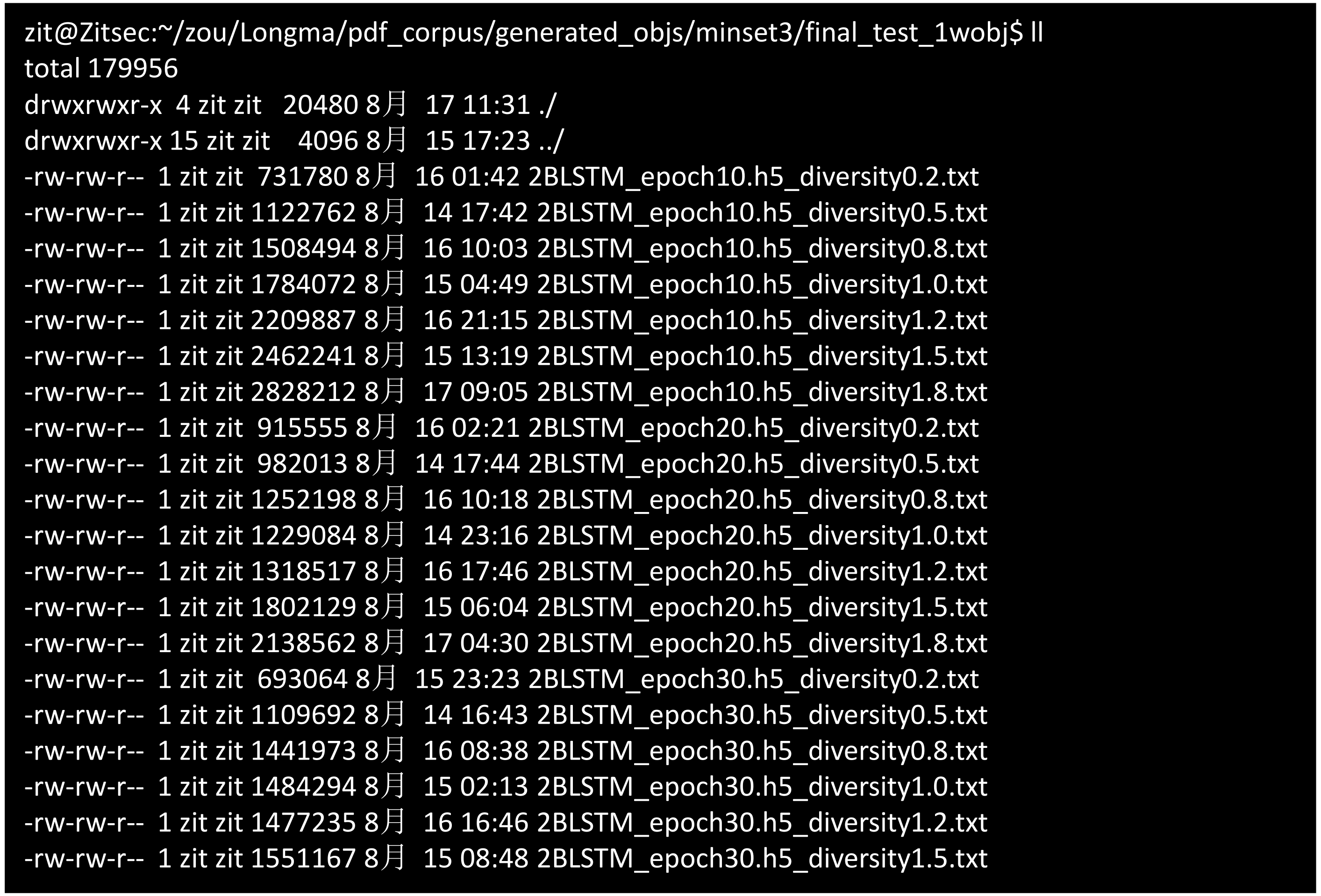
单进程生成**10,000**个obj

共计**210,000**obj

时长：**≈7小时**

单进程总时长：**7*21 = 147小时**

单个文件大小：**≈1.5MB**



```
zit@Zitsec:~/zou/Longma/pdf_corpus/generated_objs/minset3/final_test_1wobj$ ll
total 179956
drwxrwxr-x  4 zit zit   20480 8月  17 11:31 ./
drwxrwxr-x 15 zit zit    4096 8月  15 17:23 ../
-rw-rw-r--  1 zit zit  731780 8月  16 01:42 2BLSTM_epoch10.h5_diversity0.2.txt
-rw-rw-r--  1 zit zit 1122762 8月  14 17:42 2BLSTM_epoch10.h5_diversity0.5.txt
-rw-rw-r--  1 zit zit 1508494 8月  16 10:03 2BLSTM_epoch10.h5_diversity0.8.txt
-rw-rw-r--  1 zit zit 1784072 8月  15 04:49 2BLSTM_epoch10.h5_diversity1.0.txt
-rw-rw-r--  1 zit zit 2209887 8月  16 21:15 2BLSTM_epoch10.h5_diversity1.2.txt
-rw-rw-r--  1 zit zit 2462241 8月  15 13:19 2BLSTM_epoch10.h5_diversity1.5.txt
-rw-rw-r--  1 zit zit 2828212 8月  17 09:05 2BLSTM_epoch10.h5_diversity1.8.txt
-rw-rw-r--  1 zit zit  915555 8月  16 02:21 2BLSTM_epoch20.h5_diversity0.2.txt
-rw-rw-r--  1 zit zit  982013 8月  14 17:44 2BLSTM_epoch20.h5_diversity0.5.txt
-rw-rw-r--  1 zit zit 1252198 8月  16 10:18 2BLSTM_epoch20.h5_diversity0.8.txt
-rw-rw-r--  1 zit zit 1229084 8月  14 23:16 2BLSTM_epoch20.h5_diversity1.0.txt
-rw-rw-r--  1 zit zit 1318517 8月  16 17:46 2BLSTM_epoch20.h5_diversity1.2.txt
-rw-rw-r--  1 zit zit 1802129 8月  15 06:04 2BLSTM_epoch20.h5_diversity1.5.txt
-rw-rw-r--  1 zit zit 2138562 8月  17 04:30 2BLSTM_epoch20.h5_diversity1.8.txt
-rw-rw-r--  1 zit zit  693064 8月  15 23:23 2BLSTM_epoch30.h5_diversity0.2.txt
-rw-rw-r--  1 zit zit 1109692 8月  14 16:43 2BLSTM_epoch30.h5_diversity0.5.txt
-rw-rw-r--  1 zit zit 1441973 8月  16 08:38 2BLSTM_epoch30.h5_diversity0.8.txt
-rw-rw-r--  1 zit zit 1484294 8月  15 02:13 2BLSTM_epoch30.h5_diversity1.0.txt
-rw-rw-r--  1 zit zit 1477235 8月  16 16:46 2BLSTM_epoch30.h5_diversity1.2.txt
-rw-rw-r--  1 zit zit 1551167 8月  15 08:48 2BLSTM_epoch30.h5_diversity1.5.txt
```

PDF生成结果分析

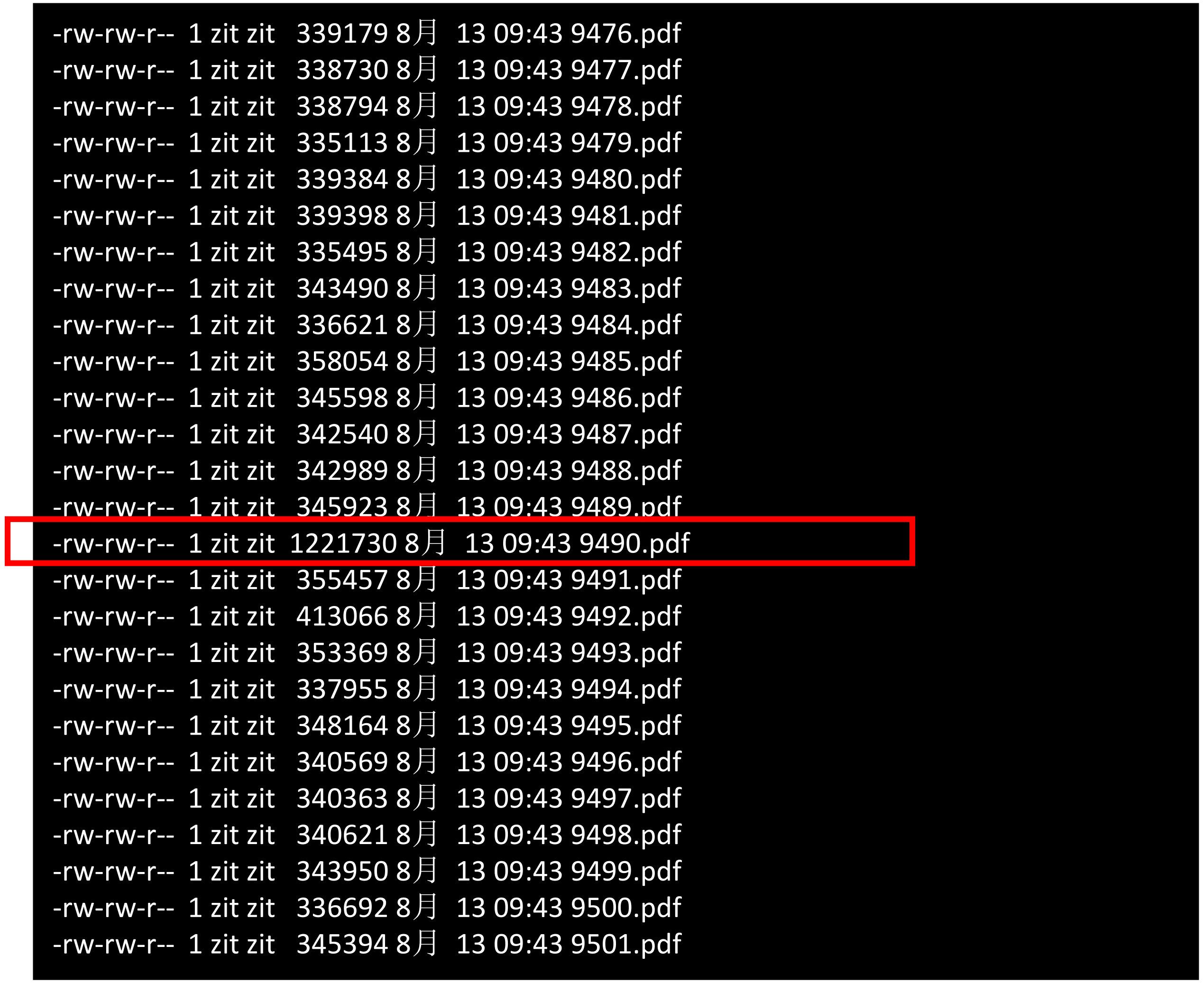
单进程生成**10,000**个PDF

时长：**≈10min**

单个大小：**≈380KB**

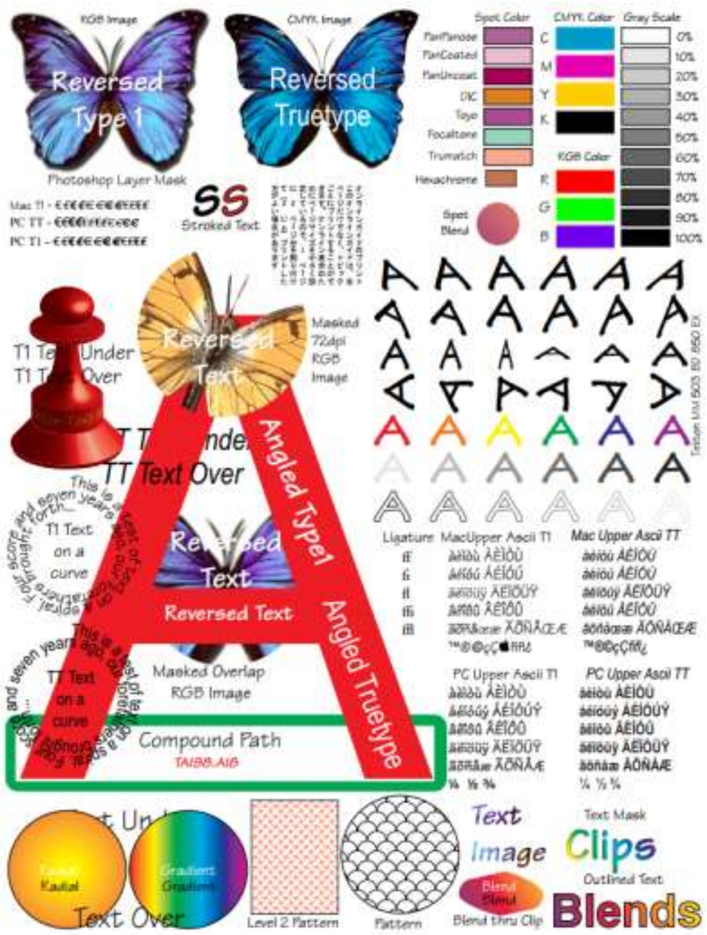
1w个文件大小：**≈3.7GB**

21个模型，共计**21w** 样本，共**77.7G**

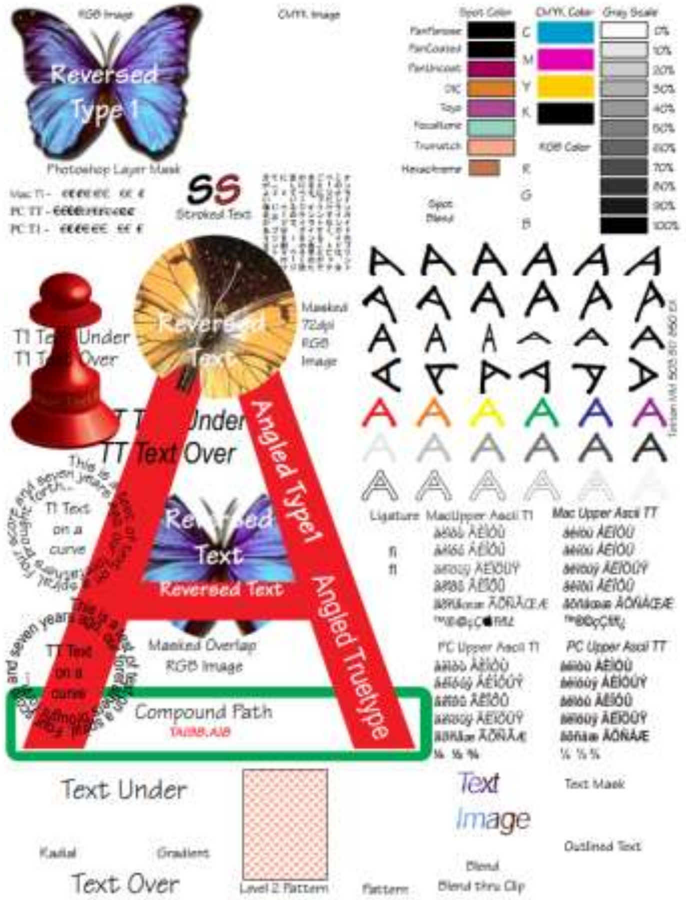


```
-rw-rw-r--  1 zit zit  339179 8月  13 09:43 9476.pdf
-rw-rw-r--  1 zit zit  338730 8月  13 09:43 9477.pdf
-rw-rw-r--  1 zit zit  338794 8月  13 09:43 9478.pdf
-rw-rw-r--  1 zit zit  335113 8月  13 09:43 9479.pdf
-rw-rw-r--  1 zit zit  339384 8月  13 09:43 9480.pdf
-rw-rw-r--  1 zit zit  339398 8月  13 09:43 9481.pdf
-rw-rw-r--  1 zit zit  335495 8月  13 09:43 9482.pdf
-rw-rw-r--  1 zit zit  343490 8月  13 09:43 9483.pdf
-rw-rw-r--  1 zit zit  336621 8月  13 09:43 9484.pdf
-rw-rw-r--  1 zit zit  358054 8月  13 09:43 9485.pdf
-rw-rw-r--  1 zit zit  345598 8月  13 09:43 9486.pdf
-rw-rw-r--  1 zit zit  342540 8月  13 09:43 9487.pdf
-rw-rw-r--  1 zit zit  342989 8月  13 09:43 9488.pdf
-rw-rw-r--  1 zit zit  345923 8月  13 09:43 9489.pdf
-rw-rw-r--  1 zit zit 1221730 8月  13 09:43 9490.pdf
-rw-rw-r--  1 zit zit  355457 8月  13 09:43 9491.pdf
-rw-rw-r--  1 zit zit  413066 8月  13 09:43 9492.pdf
-rw-rw-r--  1 zit zit  353369 8月  13 09:43 9493.pdf
-rw-rw-r--  1 zit zit  337955 8月  13 09:43 9494.pdf
-rw-rw-r--  1 zit zit  348164 8月  13 09:43 9495.pdf
-rw-rw-r--  1 zit zit  340569 8月  13 09:43 9496.pdf
-rw-rw-r--  1 zit zit  340363 8月  13 09:43 9497.pdf
-rw-rw-r--  1 zit zit  340621 8月  13 09:43 9498.pdf
-rw-rw-r--  1 zit zit  343950 8月  13 09:43 9499.pdf
-rw-rw-r--  1 zit zit  336692 8月  13 09:43 9500.pdf
-rw-rw-r--  1 zit zit  345394 8月  13 09:43 9501.pdf
```

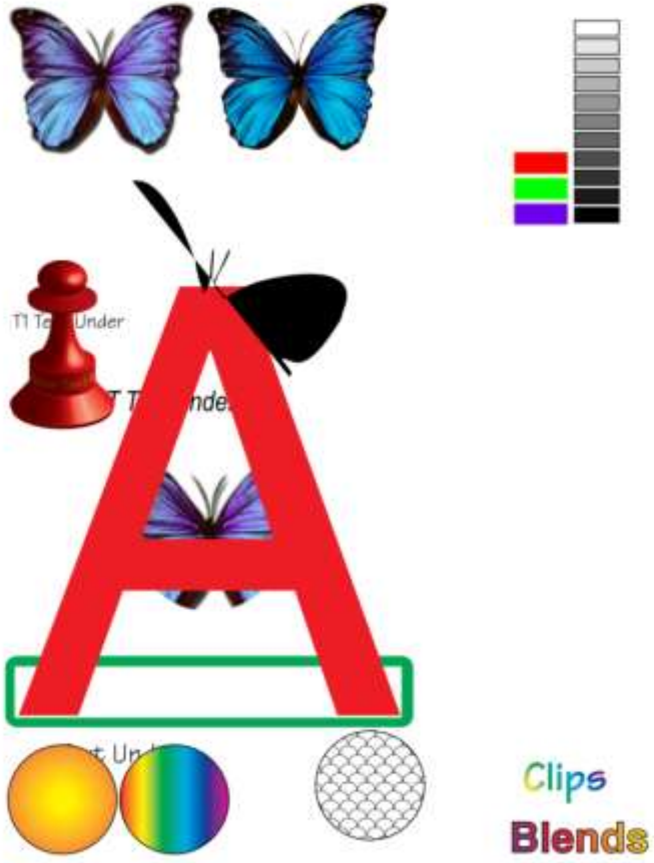
PDF生成样本示例



HOST

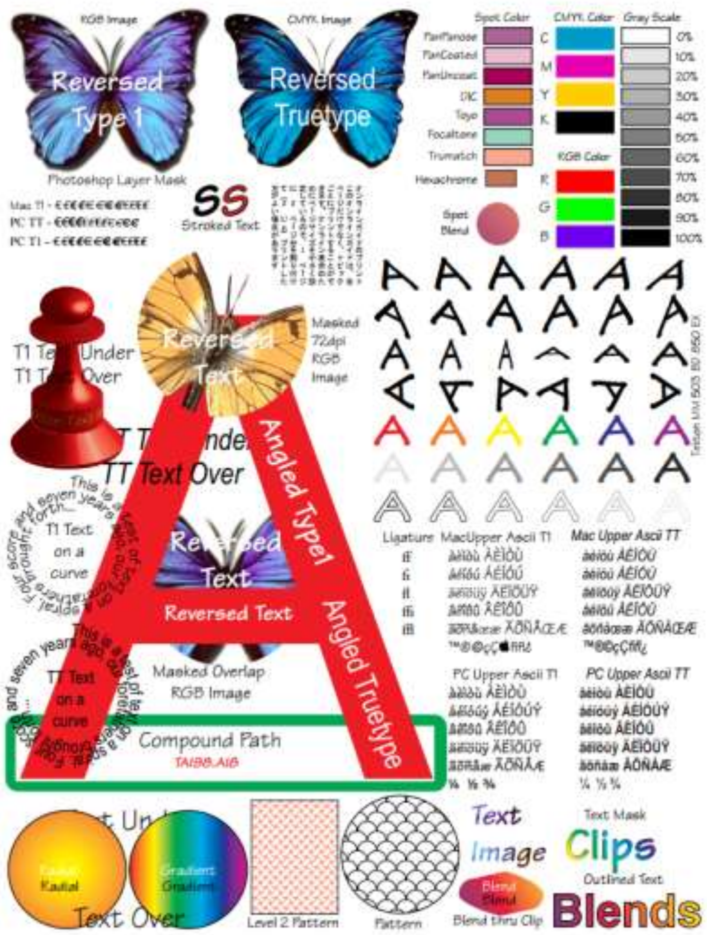


生成样本1

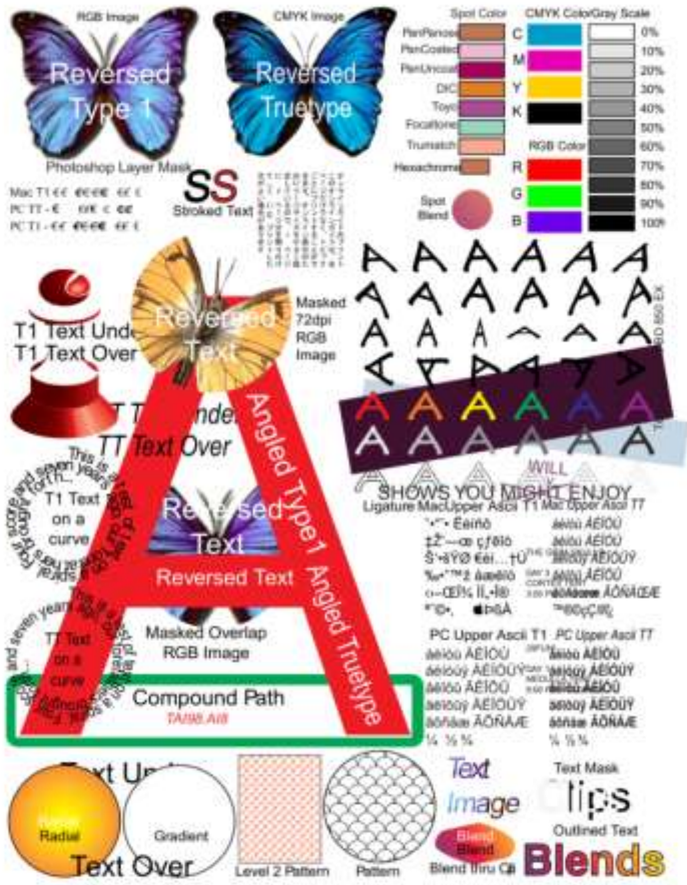


生成样本2

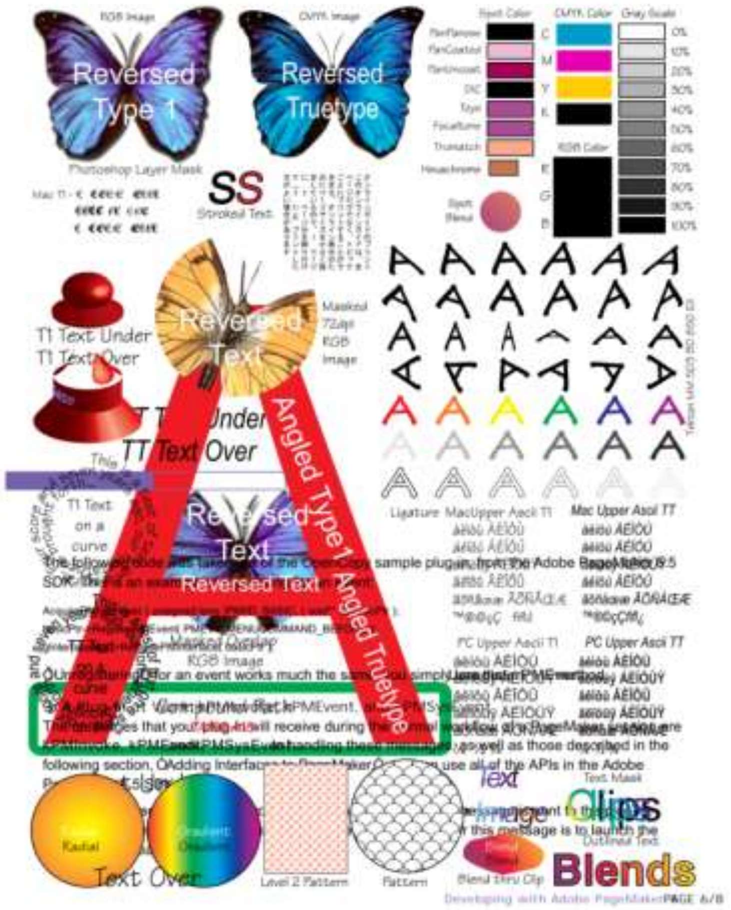
PDF生成样本示例



HOST



生成样本3



生成样本4

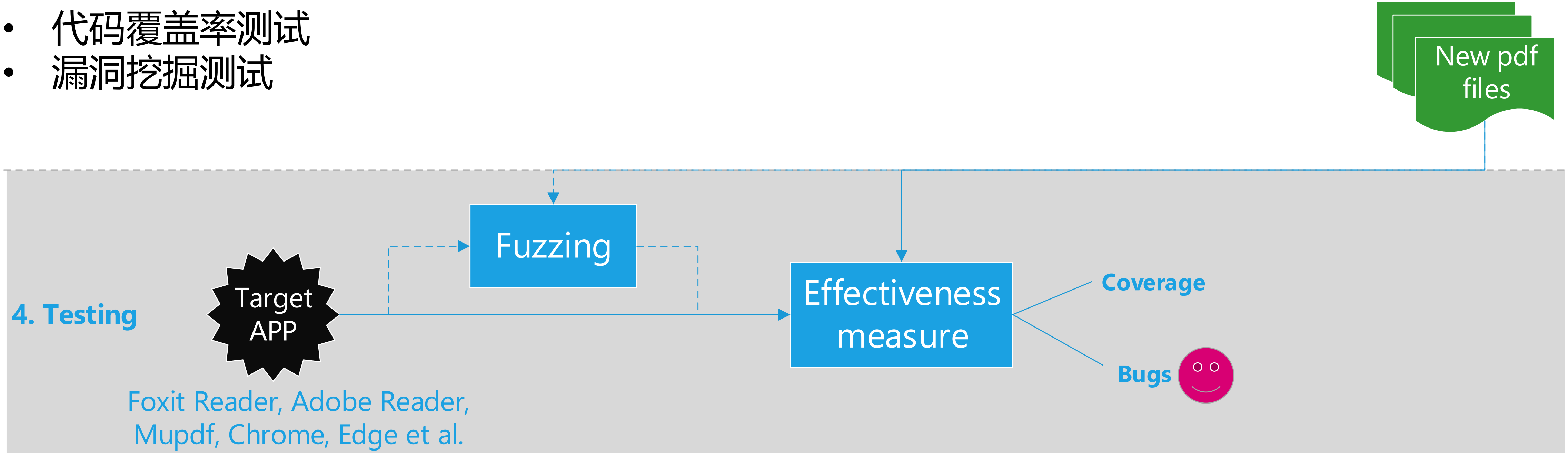
PDF样本测试

代码覆盖率测试

漏洞挖掘测试

测试（TESTING）

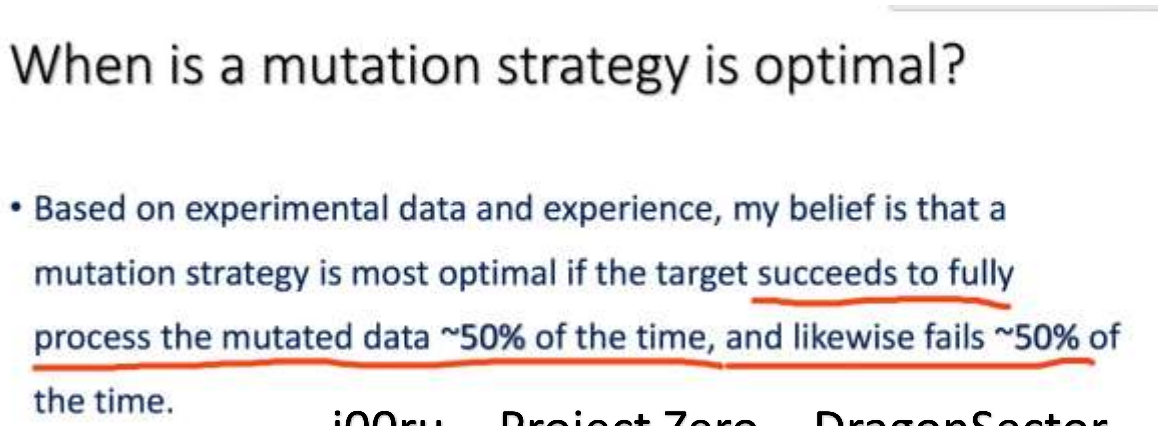
- 代码覆盖率测试
- 漏洞挖掘测试



代码覆盖率测试

代码覆盖率是评估样本质量的较好的量化指标！

微软还采用了通过率作为评估参数之一，通过率仅能反映所生成样本是否符合既定的格式规约，而代码覆盖率则能直接反映样本是否能探索到更多的路径或代码，对于漏洞挖掘具有较好的指示作用。



When is a mutation strategy is optimal?

- Based on experimental data and experience, my belief is that a mutation strategy is most optimal if the target succeeds to fully process the mutated data ~50% of the time, and likewise fails ~50% of the time.

—j00ru， Project Zero， DragonSector

代码覆盖率 = SUM（程序执行代码） / 程序总代码

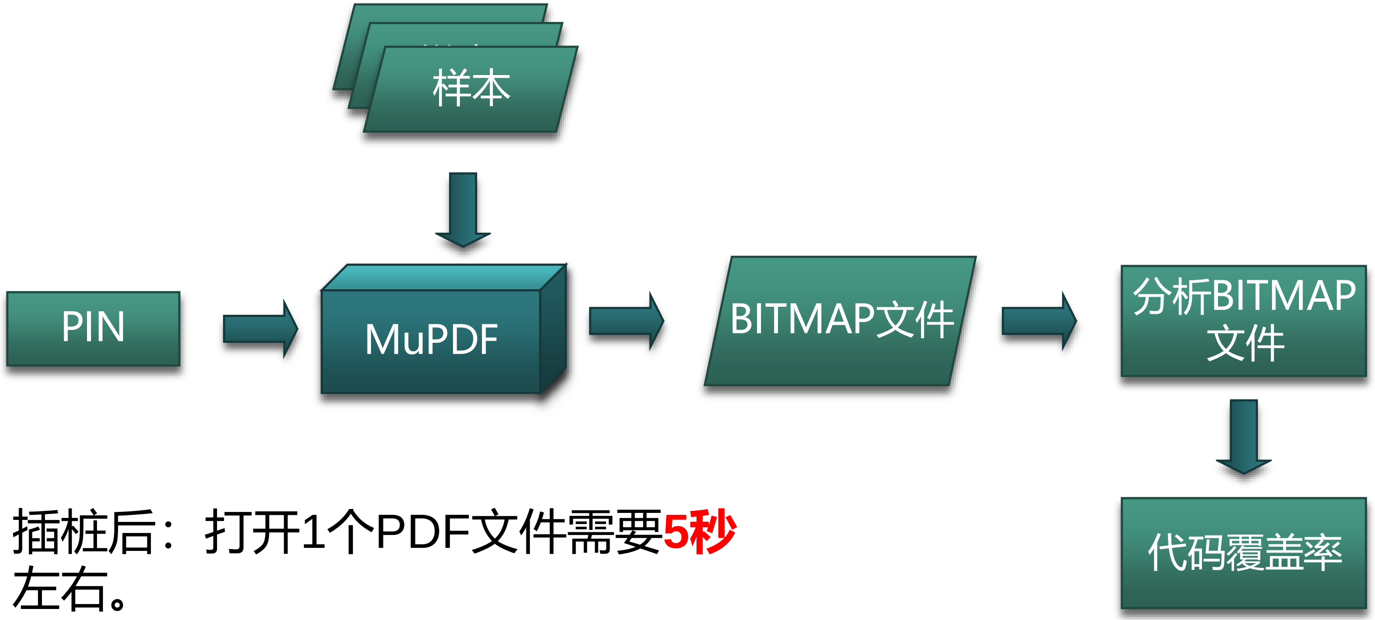
最小集后的样本数	代码覆盖率
251	**37.996%**

采用MuPDF作为测试代码覆盖率的载体

WHY MUPDF?

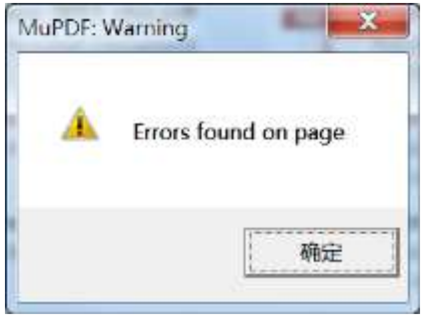
- 静态链接，所有库all in one file
- 功能全，支持各种形式stream
- 轻量级，易插桩
- 几乎无bug，测试数据更准确
- Open source ,易分析

代码覆盖率测试



插桩后：打开1个PDF文件需要**5秒**左右。

NOTES：对于有些弹框需模拟点击，以使样本能充分测试。

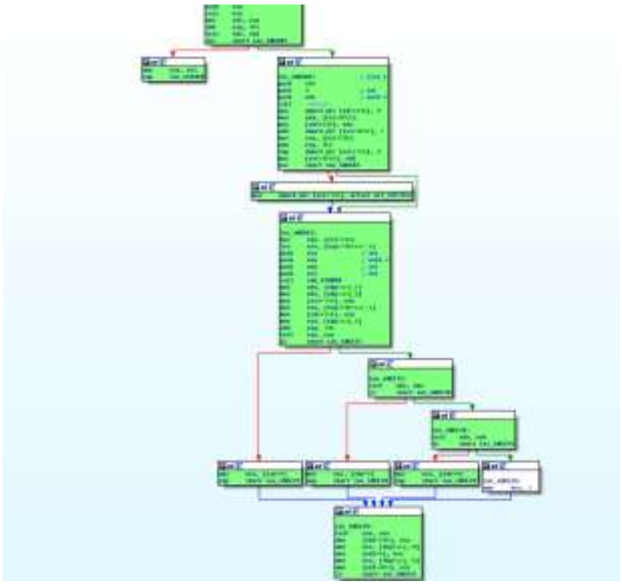

MuPDF: Warning

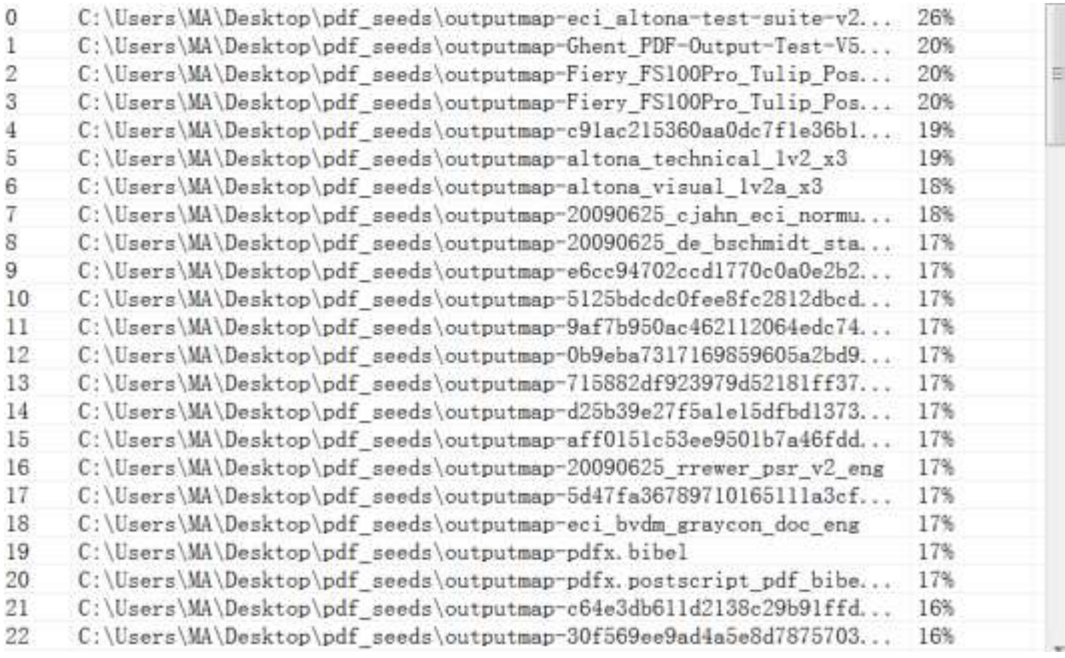
Errors found on page

PIN作为商业的轻量级插桩工具，具有较好的性能和稳定性表现。

优化：

- ✓ 基本块级插桩
- ✓ CPUKill
- ✓ 1bit 表示1 Byte, Zlib压缩



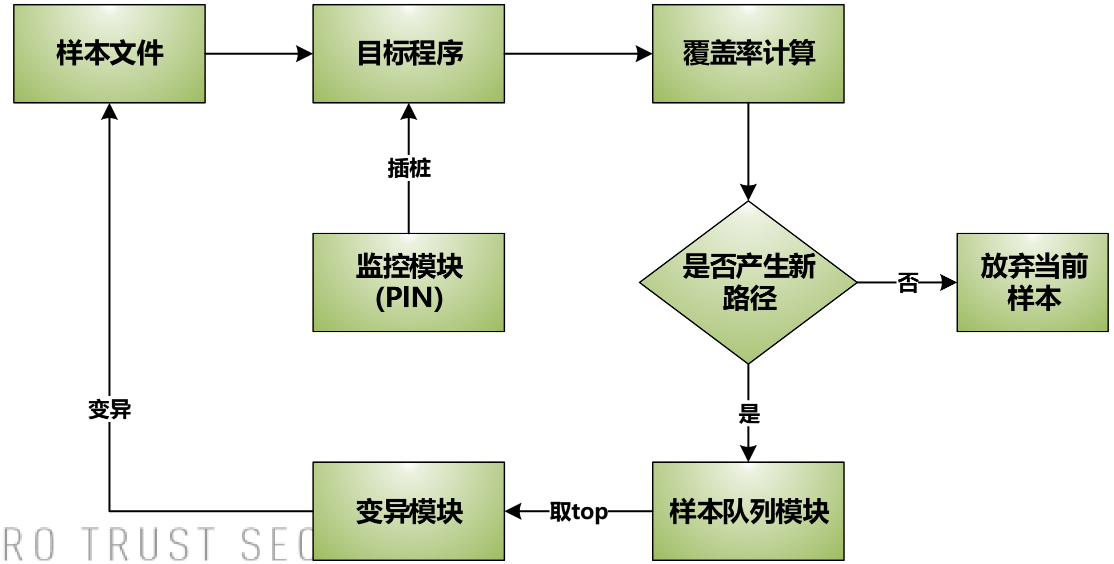


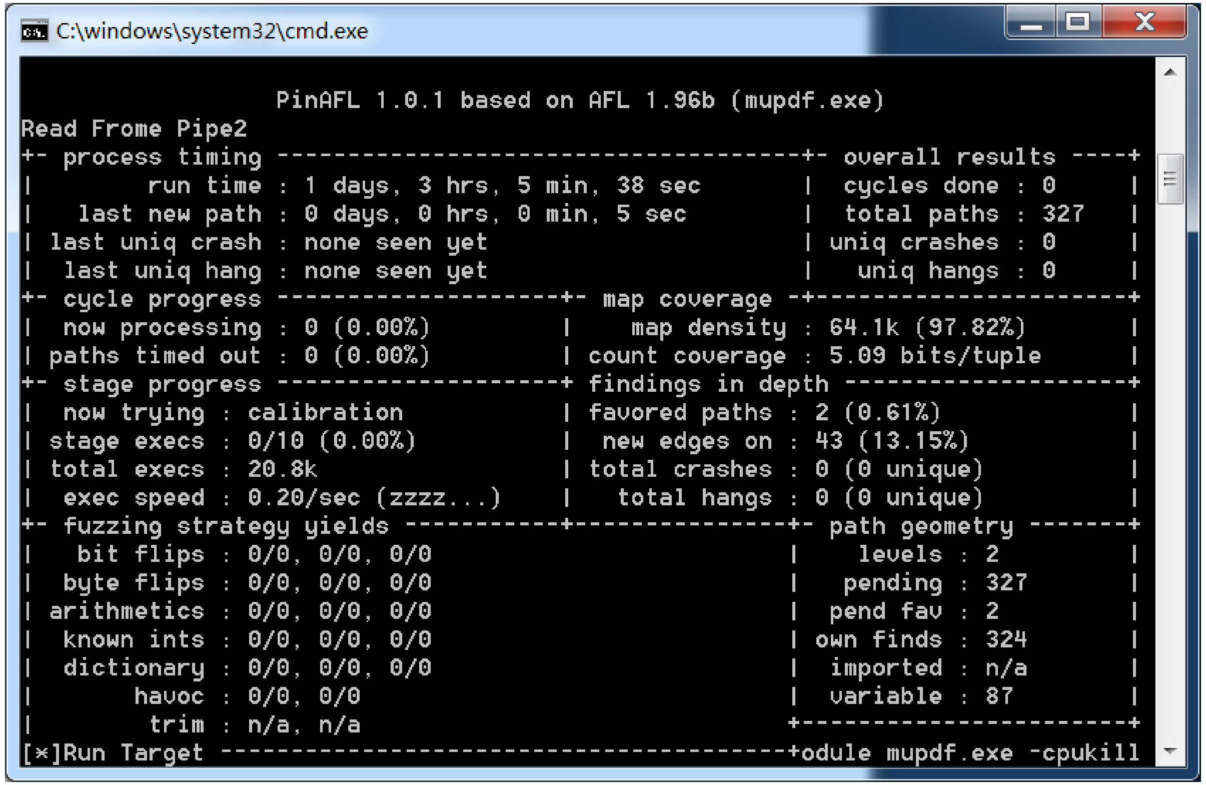
0	C:\Users\MA\Desktop\pdf_seeds\outputmap-eci_altona-test-suite-v2...	26%
1	C:\Users\MA\Desktop\pdf_seeds\outputmap-Ghent_PDF-Output-Test-V5...	20%
2	C:\Users\MA\Desktop\pdf_seeds\outputmap-Fiery_FS100Pro_Tulip_Pos...	20%
3	C:\Users\MA\Desktop\pdf_seeds\outputmap-Fiery_FS100Pro_Tulip_Pos...	20%
4	C:\Users\MA\Desktop\pdf_seeds\outputmap-c91ac215360aa0dc7f1e36b1...	19%
5	C:\Users\MA\Desktop\pdf_seeds\outputmap-altona_technical_1v2_x3	19%
6	C:\Users\MA\Desktop\pdf_seeds\outputmap-altona_visual_1v2a_x3	18%
7	C:\Users\MA\Desktop\pdf_seeds\outputmap-20090625_cjahn_eci_normu...	18%
8	C:\Users\MA\Desktop\pdf_seeds\outputmap-20090625_de_bschmidt_sta...	17%
9	C:\Users\MA\Desktop\pdf_seeds\outputmap-e6cc94702ccd1770c0a0e2b2...	17%
10	C:\Users\MA\Desktop\pdf_seeds\outputmap-5125bdcdc0fee8fc2812dbcd...	17%
11	C:\Users\MA\Desktop\pdf_seeds\outputmap-9af7b950ac462112064edc74...	17%
12	C:\Users\MA\Desktop\pdf_seeds\outputmap-0b9eba7317169859605a2bd9...	17%
13	C:\Users\MA\Desktop\pdf_seeds\outputmap-715882df923979d52181ff37...	17%
14	C:\Users\MA\Desktop\pdf_seeds\outputmap-d25b39e27f5a1e15dfbd1373...	17%
15	C:\Users\MA\Desktop\pdf_seeds\outputmap-aff0151c53ee9501b7a46fdd...	17%
16	C:\Users\MA\Desktop\pdf_seeds\outputmap-20090625_rrewer_psr_v2_eng	17%
17	C:\Users\MA\Desktop\pdf_seeds\outputmap-5d47fa36789710165111a3cf...	17%
18	C:\Users\MA\Desktop\pdf_seeds\outputmap-eci_bvdm_graycon_doc_eng	17%
19	C:\Users\MA\Desktop\pdf_seeds\outputmap-pdfx.bibel	17%
20	C:\Users\MA\Desktop\pdf_seeds\outputmap-pdfx.postscript_pdf_bibe...	17%
21	C:\Users\MA\Desktop\pdf_seeds\outputmap-c64e3db611d2138c29b91ffd...	16%
22	C:\Users\MA\Desktop\pdf_seeds\outputmap-30f569ee9ad4a5e8d7875703...	16%

代码覆盖率测试

PINAFL — 基于PIN实现了AFL的WINDOWS版本

- 运行了**1天3小时**
- **20,000**多次变异
- 发现了**327**条新的路径，即产生了**327**个新的测试用例。
- 代码覆盖率为：**38.077%**





```
                  PinAFL 1.0.1 based on AFL 1.96b (mupdf.exe)
Read Frome Pipe2
+- process timing -------------------------------------+- overall results ----+
|        run time : 1 days, 3 hrs, 5 min, 38 sec       |  cycles done : 0     |
|   last new path : 0 days, 0 hrs, 0 min, 5 sec        |  total paths : 327   |
| last uniq crash : none seen yet                      | uniq crashes : 0     |
|  last uniq hang : none seen yet                      |   uniq hangs : 0     |
+- cycle progress --------------------+- map coverage -+----------------------+
|  now processing : 0 (0.00%)         |    map density : 64.1k (97.82%)       |
| paths timed out : 0 (0.00%)         | count coverage : 5.09 bits/tuple      |
+- stage progress --------------------+ findings in depth --------------------+
|  now trying : calibration           | favored paths : 2 (0.61%)             |
| stage execs : 0/10 (0.00%)          |  new edges on : 43 (13.15%)           |
| total execs : 20.8k                 | total crashes : 0 (0 unique)          |
|  exec speed : 0.20/sec (zzzz...)    |   total hangs : 0 (0 unique)          |
+- fuzzing strategy yields -----------+---------------+- path geometry -------+
|   bit flips : 0/0, 0/0, 0/0                         |    levels : 2         |
|  byte flips : 0/0, 0/0, 0/0                         |   pending : 327       |
| arithmetics : 0/0, 0/0, 0/0                         |  pend fav : 2         |
|  known ints : 0/0, 0/0, 0/0                         | own finds : 324       |
|  dictionary : 0/0, 0/0, 0/0                         |  imported : n/a       |
|       havoc : 0/0, 0/0                              |  variable : 87        |
|        trim : n/a, n/a                              +-----------------------+
[*]Run Target ----------------------------------------+odule mupdf.exe -cpukill
```

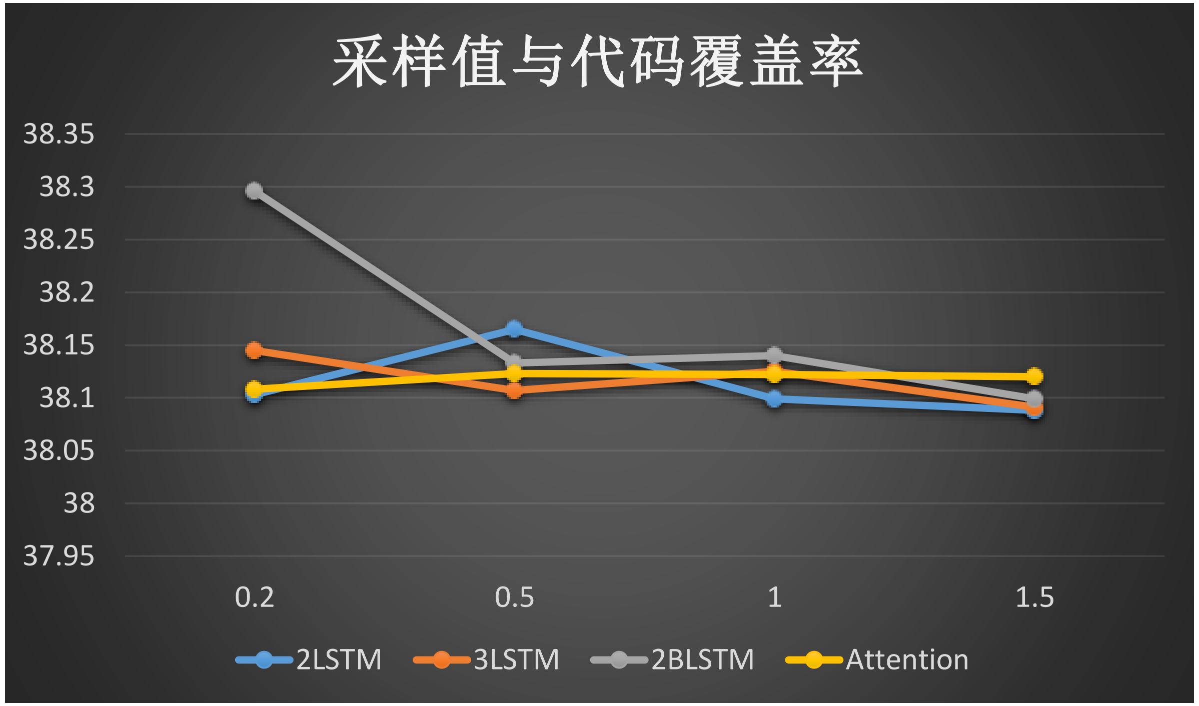
未修改AFL的变异算法和调度算法，因此能较真实体现AFL的水平

结果分析

采样值对代码覆盖率的影响

轮次：**60**
测试时长：**13.89*16=222.24小时**
代码覆盖率最高提升**0.3%**，约**20,000+**指令

	2LSTM	3LSTM	2BLSTM	Attention
0.2	38.103	38.145	38.296	38.108
0.5	38.165	38.107	38.133	38.123
1.0	38.099	38.125	38.140	38.122
1.5	38.088	38.091	38.099	38.120



数据集基础覆盖率：**37.996%**
PinAFL覆盖率：**38.077%，+0.081%**
Learn&Fuzz覆盖率：**38.113%，+0.117%**

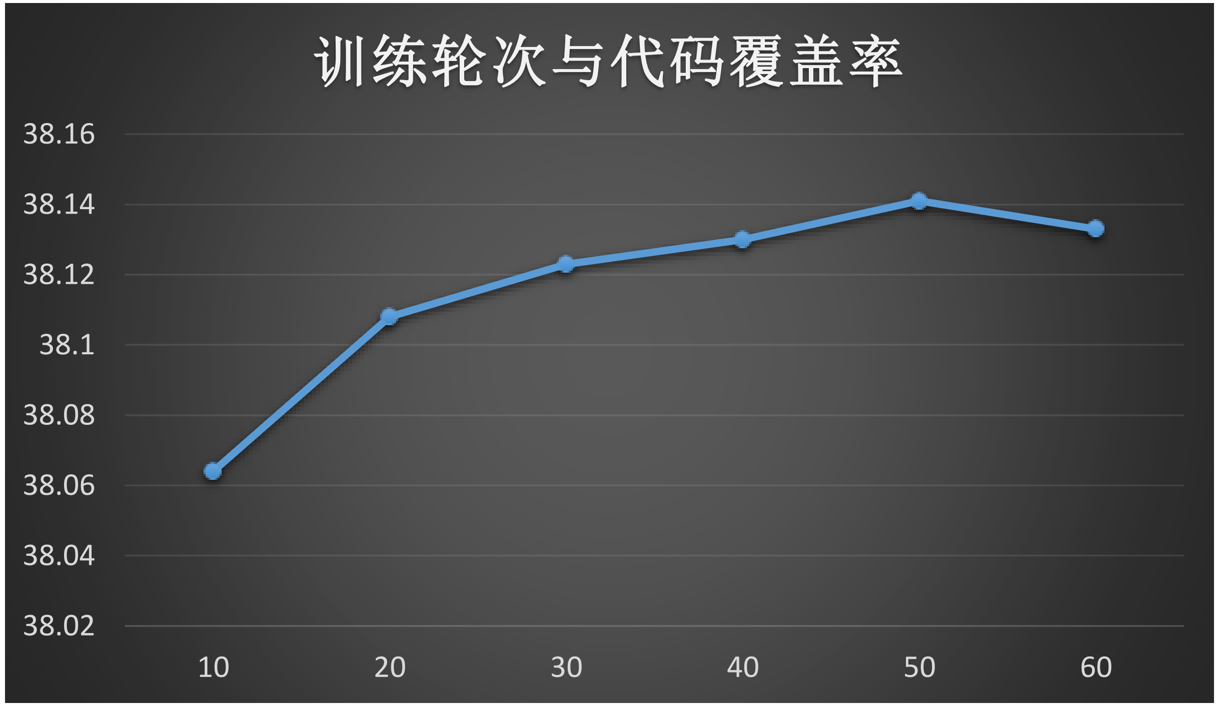
结果分析

ISC 2018 ISC互联网安全大会 360互联网安全中心

训练轮次对代码覆盖率的影响

模型：**2BLSTM**
采样值：**0.5**
测试时长：**13.89*5= 69.45小时**

轮次	代码覆盖率
10	38.064
20	38.108
30	38.123
40	38.130
50	38.141
60	38.133



漏洞挖掘测试

采用我们的方案生成的PDF文件，对**Foxit Reader**、**Power PDF**、**Corel PDF**、**Cool PDF**、 **Nitro PDF**等软件进行了测试。

采用**集群漏洞分析系统**作为测试平台，分别为每个测试对象分配了**20台**虚拟机，测试时间为**1天**，测试样本数为：**210,000**

结果分析

软件名	crash数量	去重后	漏洞类型
powerPDF	4520	28	TaintedDataControlsWriteAddress、StackOverflow、TaintedDataControlsBranchSelection、ReadAVonControlFlow、TaintedDataControlsBranchSelection等
corelPDF	23560	78	WriteAV、ReadAV、TaintedDataControlsBranchSelection、DivideByZero等
coolPDF	468	8	TaintedDataReturnedFromFunction、TaintedDataControlsWriteAddress、ReadAVNearNull等
Nitropdf Reader	256	5	TaintedDataControlsBranchSelection、TaintedDataPassedToFunction等
Foxit92	10265	27	TaintedDataControlsCodeFlow、ReadAV、DivideByZero、StackOverflow等
Foxit91	2783	18	TaintedDataPassedToFunction、TaintedDataReturnedFromFunction、StackOverflow等
总数		164	

其中某个漏洞已经被判定为可利用!

结论与展望

结论

展望

结论

1. 本方案实现了一种基于AI制导的PDF文件生成技术，方案具有以下特性：
   - 支持**Char-level**的学习
   - 支持**LSTM、BLSTM、Attention**机制网络模型
   - 支持基于离线模型、字典和多采样值的obj生成（进程池）
   - 支持基于离线obj文件的PDF样本生成（进程池）

2. 对不同模型及不同参数进行了较严谨的测试，在本次测试中，**高训练轮次、低采样值**生成的样本具有更高的代码覆盖率，其中**2BLSTM**模型**60**轮采样值**0.2**的表现效果最佳；

3. 本方案可落地实现为一种新的样本变异策略，可单独生成样本用于漏洞挖掘，也可作为AFL等工具的前端，但还不能完全取代当前主流Fuzzer。

展望

1. 支持更多的结构化样本格式的学习和生成，如**XML、XSL、JavaScript、HTML、 AS**等
2. 训练二进制格式(**PNG、MKV、ZIP**等)，看是否能生成较通用的模型。**难点：**校验和、二进制规律性不强
3. 把生成的样本交给AFL进行Fuzzing，看能否增强AFL本身的性能；
4. 单一模型与多模型组合比对
5. 交互方式训练模型：**GAN**

参考资源

- Adobe Systems Incorporated. PDF Reference, 6th edition, Nov.2006. Available at http://www.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/pdf reference 1-7.pdf
- Wang J, Chen B, Wei L, et al. Skyfire: Data-driven seed generation for fuzzing. Security and Privacy (SP), 2017 IEEE Symposium on. IEEE, 2017: 579-594.
- https://patricegodefroid.github.io/
- https://patricegodefroid.github.io/public_psfiles/SAGE-in-1slide-for-PLDI2013.pdf
- Godefroid P, Peleg H, Singh R. Learn&fuzz: Machine learning for input fuzzing. Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering. IEEE Press, 2017: 50-59.
- https://github.com/keras-team/keras/blob/master/examples/lstm_text_generation.py
- https://github.com/philipperemy/keras-attention-mechanism

ISC 2018
ISC互联网安全大会

360互联网安全中心

谢 谢！

2018 ISC 互联网安全大会 中国 · 北京
Internet Security Conference 2018 Beijing · China
（原中国互联网安全大会）