

如何7天写出一篇会议论文

周俊宇

2024年4月25日

2024-04-09 ~ 2024-04-16

(当然是在实验做完的情况下)

事情是这样的：

我大概从这学期开始完成我工作的实验，每天早上十点到靶场，晚上十点回寝室。除了周二周四周六会在晚上八九点跑步三公里外，其余时间基本上就是：

一台盖着的笔记本、一台显示器，以及——几十份不可维护的、数千行的代码和祖传的数据集。

```
~/Desktop/old_desktop/scu/research/2024/src @006ef551 *2 !2 ?1 13:14:25
> ls
├── gycache
├── accuracy_lab2
├── accuracy_result
├── accuracy_result.best
├── calculate_accuracy.py
├── calculate_accuracy_patch.py
├── calculate_accuracy_with_graph.py
├── call_graph.png
├── evaluate.py
├── extract_report_ttp.py
├── host_get_stacktrace.sh
├── host_get_stacktrace_batch.bak
├── host_get_stacktrace_batch.sh
├── host_patch.sh
├── host_patch_batch.py
├── inv_mapping
├── inv_mapping_additional
├── inverse_mapping.py
├── ispackaged.pyc
├── locate.py
├── map_from_line_to_file.txt
├── mapping
├── mapping_additional
├── output_folder
├── patch.py
├── patch_pure.py
├── performance_run
├── post_evaluate.py
├── report.json
├── stktrace_add_func.py
├── stktrace_output
├── stktrace_output_new
├── test.exe
├── test.exe.patched
├── text2dict.py
├── ttp_from_stacktrace_per_line.txt
├── ttp_from_stacktrace_per_line_new.txt
├── ttp_from_stacktrace_per_line_new1.txt
├── ttp_one_file_per_line.txt
├── ttp_one_file_per_line_new.txt
├── updated_data.json
└── VirusShare_0aa06ef1402a0c32b1f00003f0b53f54a1

~/Desktop/old_desktop/scu/research/2024/src @006ef551 *2 !2 ?1 13:14:26
> find -type f \( -name "*.py" -o -name "*.sh" \) -exec wc -l {} \; | awk '{total += $1} END {print total}'
2173
```

在起早贪黑，废寝忘食地努力下，我成功在3月底做完了两个比较大的实验，结果也还尚可，有了发论文的实验基础。

于是，正题开始了

得知会议

我本来是刚刚把实验做完就让导师给我找一下适合投的会议，我说我想在四月份投稿，结果过了一周（也就是4月9号早上），他才给我找了 RAID^[1] (CCF-B)。

上会伴一看^[2]，这个会议的ddl还有10个小时？点进去一看，哦，主办方把ddl从4月9号延期了一周到北京时间4月16号晚上8点，我松了一口气。

我当时一脸问号——导师啊，但凡你要是提前一周告诉我要投这个会议，我的现状就不会像今天这样狼狈。

结果导师说，你先花个两天把论文写出来，然后我们再改。

。 。 。 。 。 。

1. <https://raid2024.github.io/> ↩

2. <https://myhuiban.com/conferences/host> ↩

寻找环境

不巧的是，由于靶场那段时间要进行装修，我在靶场的工位不保，得另寻它处。

寝室肯定是不行的，虽然亲爱的毛主席为了磨练意志要到闹市学习，然而他并不需要为一个七天的ddl而奋斗。

图书馆的话我有所考虑，但是相比之下高凯老师的实验室会更好些，因为运气好的话可以问身边的学长学姐一些问题。

于是我在高凯老师的同意下，在当天下午把家当全搬进了实验室的最后一排。

网安楼虽然不大，但已经能安放一张平静的书桌了。

此时我打开投稿网站，确认要求是 double column 的 ACM sigconf style，于是我熟练地打开了 overleaf^[1] 找到了对应的模板。

然后我又意外地发现，由于去年在NUS写过project，我的overleaf账号仍然绑定着NUS的临时邮箱，成为了overleaf的会员。因此，团队协作overleaf写论文是可行的。

1. <https://www.overleaf.com/> ↩

确定idea没人做过

由于整个三月份我在兢兢业业地做着实验，只是因为凭着导师“这个工作以前没人做过”这句话，并没有进行实地的文献搜集。当我想找些人手和我一起写论文时，李梓勤同学就问我这个工作应该有人做过吧，我说我不知道，但导师说没有，我也就没想那么多了。

Ziqin Li: 。 。 。 。 。

Ziqin Li: 你今天赶紧去搜一下有没有相关的工作——immediately！！

我其实内心也有点后怕，要是这个工作真被做过，那我不就成纯纯大冤种了吗，白花了一个月宝贵的时间。

于是，我在高老师的实验室问了同桌一个问题：如何去找相关的工作？

那个学长问我你的导师有没有给你发过相关的工作，哪怕给一篇也行，我说我有学姐的一篇《川大学学报》。

学长说那行，你把她的参考文献里面跟你工作最相近的那篇文章拿出来，放到 `connected papers`^[1] 里面一查，再看一下那里面相关的工作，就基本上能确认你的工作有没有被做过了。

事实证明导师没有骗我，我的方向确实只有一两篇paper做过，方法也不同，而且发的会议都不在CCF里面。

1. <https://www.connectedpapers.com/> ↩

开写

所以，先写什么呢？

按照兰晓老师的说法，建议先写background、methodology和evaluation，然后再进行discussion和related work，再之写intro和conclusion，最后补上abstract。

我觉得她的说法很有道理，毕竟一个现在才补related work的ddl人，background肯定是最好写的。

因为是现有的基础知识，所以用chatgpt的话应该没什么大问题，然后再加点与我自己工作的关系和相关的feature应该就可以。

于是，我花了周二一下午就把background给写完了。

接着是methodology，由于这部分就是自己的思路了，因此让chatgpt愉快地帮我生成内容的时光到此结束了。

导师说methodology最好有一个项目的整体架构图，我灵机一动，之前信安赛做ppt的那位同学画的那个架构图似乎很不错。于是我让他把架构图的中文改成英文，贴上去一看，挺match的。

然后methodology的话我有很多算法的部分，然而 LaTeX 只有在用 markdown 写博客的时候零零星星敲几个公式时会用到，因此这方面还是得求助chatgpt给我生成一个算法的模板，chatgpt用了algorithm2e，感觉效果还行。我多看了几个例子就掌握了它的大致用法。

不得不说算法真是撑篇幅的有力工具——

然后具体的文字叙述就只能靠自己对工作的细节回忆了，这方面确实比较痛苦，直到晚上实验室关门时我才写完了我工作的一半内容。

直到周三结束，我才完成了methodology这一部分——所以两天是写不完一篇paper滴！

补做实验

先前李梓勤同学也跟我argue说你的论文还缺少两个点，一个是motivation（也就是你做这个工作的意义何在），另一个就是实验太不充分了。他说如果你不做性能实验，审稿人就会怀疑你的工作耗时太长，想要刻意隐瞒（效率低）这件事情。

我当时想了想，性能实验确实不太难做，在我一边构思论文措辞时一边跑实验是完全可行的。另外，由于相关工作大多数都是用了机器学习相关的方法，缺乏可解释性。因此我也补做了一个简单的可解释性实验。

画图的话也只能靠 chatgpt，我大概给它描述一下怎么给数据分组、然后画条形图还是折线图、颜色、图例、注释等等，然后在它给的代码基础上魔改，感觉确实比读 matplotlib 的文档快一些。

在做实验的同时，我把我的工作介绍给了一些关系比较好的同学，求他们帮我写一下 related work 和 intro。我大概在周五的时候补完了我能补的实验并写好了实验部分。周六写discussion的时候有点颓（其实可能是没啥思路），我就直接把related work介绍自己工作的优点的那一部分移过去了。

然后那周日催一催那位同学把 related work 和 intro 写了，然后支棱了一个下午晚上补全了conclusion和abstract，用overleaf copilot^[1]修改了一下语病，这篇论文的初稿就算完成了。

1. <https://www.overleafcopilot.com/> ↩

修改的噩梦

由于周日晚上我把论文初稿发在了我临时组建的写论文的群里，导师在第二天提出了一些论文的美化意见：例如说图的风格保持统一、附录的内容充实一点（最好带几张图）、论文的background部分最好不要使用定义性的语言等等，反正又忙活了一个下午加一个晚上才搞定这些修改。

不仅如此，这个时候 Ziqin Li 又突然冒出来说：

- 你论文怎么没有motivation？
- 你的实验和related work怎么这么短？而且内容与你的工作一点关系都没有？
- 还有 discussion 不是你这么写的！

请求我当时心里的阴影面积——

看来 related work 还是得自己亲手操刀啊。。。。。。

没办法，writing的问题并不是我一个人能解决的，只能和他约好ddl当天下午和晚上一起改论文——太刺激了！

我记得16号下午我们从吃完午饭开始，Ziqin Li先确认了我的工作是 ISA-independent 的，但是论文 methodology 中间 `eax` , `call, DWORD ptr` 满天飞。然后他又让我补我的算法的 benchmark，刚好那个时候代码又跑不通了。我一边在论文中极力降低自己的措辞与 ISA 的耦合性，一边疯狂调试我那边不可维护的代码，一边上网查、chatgpt，问我的那些汇编代码在不同 ISA 的体现方式，忙得不可开交。

终于，在吃晚饭之前，我完成了这两个比较重要的任务。此时，我们只剩下不到三个小时的时间了。

然后到了晚上，这个时候疯狂重写 motivation, related work 和 discussion。这个时候纯靠自己一句一句写已经来不及了，于是我选择在 motivation 部分给出一个开头，然后让 overleaf copilot 续写下面的部分。虽然它写的东西近似于流水账，但读着还是比较舒服的。然后我又在兵荒马乱之中帮 Ziqin Li 写了一部分 discussion，与此同时我也先打开了论文的提交网站，填写了必要的信息。只等他一声令下我把 PDF 传上去点击提交。Ziqin Li 在完成他的部分的同时，调整了一下论文的 structure，把 related work 放在了最后（目的就是写得太烂了不想被 reviewer 细看）。

最后10分钟，我们快速读了一遍文章，没有发现明显的排版错误和语法错误。

最后5分钟，我们再次确认论文是匿名提交的。

最后2分钟，我们上传论文后点击了提交按钮。系统显示我们是第212篇提交的论文，提交时间是2024年4月16日晚上7点58分26秒（UTC+8）。

其实ddl过的十分钟内也是可以交论文的，并不会卡得太死（但不要冒险）。故事结束。

总结

- LaTeX 从来都是现学才会用的。
- overleaf 确实适合论文协作。
- connected papers 可以快速确定你的工作有没有人做过，并为相关工作的撰写提供了方便。
- chatgpt 可以用来写常识性的部分，也可以对画图、画表、算法绘制部分带来一定帮助。（3.5就够了）
- 有一个 PPT 大师是最好的。
- 实在脑子里一团浆糊可以用 overleaf copilot帮你续写内容，同时用于语法查错也是不错的选择。
- 专业领域里自己不想写的内容不要给别人写，因为通常情况下写出来的东西会更烂。