

Structure Instructions for an Academic Paper in CS

徐悦甦/西电计算机学院

有一些同学反映，第一次写论文时，在结构上会遇到不少问题，比如一个部分写了几句就不知道再写什么；一篇文章，把自己的方法用公式和几个图交代完了就不知道还需要再交待什么；一个技术点在论文里重复好几遍，不知道是不是应该这样；实验把结果一列就结束了；为什么要写这一部分，为什么又要写那一部分等等。这里和大家分享一些关于 academic paper 的 structure 的内容。

论文的结构可以分为大结构和小结构，大结构就是下面的一级标题，比较容易，关键在于小结构，即每一部分的每一个子部分又怎么写。其中的一些部分写法并不唯一，但是对于一个 starter，可以首先参考经典的写法，当自己已经写过一篇论文后，就会有自己对于结构的认识。下面的结构比较适用于 CS 的长文，包括会议和期刊，短文和 demo 写法更灵活。

1. Title

起法灵活，归根结底是要反映论文做的核心的事情，即最可以拿出来说的那一部分解决了一个什么问题，如果场景很重要还可以加上在什么环境下。比如使用了社交信息和异构信息网络，那就是 Social XXXXX / Heterogeneous Information Network，解决了个性化推荐问题，那就是 Social Recommendation with Heterogeneous Information Network。

如果就是单纯提出了一个方法，那题目可以比较短，比如 Neural Collaborative Filtering

还有突出目的的起法，即“我们的这篇论文就是为了干这个事”，往往这种起法的背景是方法不是第一位的，问题才是第一位的，比如 Towards XXXX via XXXX

一般情况下面向应用问题的研究论文，题目总归可以起成 XXX-based/aware/enhanced/extended XXXX in/for XXX 或者”形容词 +XXXX using/via/through/with XXX and XXX”。一些论文的作者会为自己做出的工具或者提出的模型给一个缩写词，作为题目的打头，也很常见。还有一小部分可以起成问句：Can XXXX solve XXX? The answer is Yes.

2. Abstract

第一句：点出背景，即为什么会有这个问题？这个问题来自哪里？一般情况下我们的问题都是有来源的，即不是凭空想出来的，而面向应用的研究问题，来源一般是来自应用本身，即实际问题的需要。

第二句：点出解决问题的重要性/必要性/急迫性等，即不但是有这个问题，而且这个问题的解决是很重要的，或者很必要的，或者很急迫的，不解决就会影响 XXX，或者阻碍 XXX

第三句：现有的解决方法存在问题，可能是一个大的问题，也可能是一系列问题。如果这个问题很新，还没有特别对口的论文解决，那可以说：还没有论文解决，但是前提是做好调研。不要出现，其实是有论文已经解决过一次几次，结果还说：没有人解决。有的时候，第二句和第三句的内容是可以合并的。

第四句：开始起笔写 To solve these problems, To address these issues, To attack XXX，即在这篇论文里，我们总体干了一个什么。比如提出了一个需求敏感的移动拼车路线规划方案，面向移动边缘环境，提出并实现了一个 XXXXX。在这里不要太强调基于的方法，因为（特别是面向应用的研究），问题是第一位的，方法是第二位的，不管用了什么方法，都是为了解决这个问题。

第五句：开始具体写我们干的事情，按照论文正文中我们解决问题的正常顺序，比如建立了一个 XXX 模型，提出了一个 XXX 算法，建立了一个 XXX 架构，提出一个新的 XXX 技术。这时候就可以写基于什么方法或者算法。

第六句：如果干的具体的事比较多，比如提出了两个模型，或者提出一个模型，也提出了一个算法，或者设计一个神经网络架构，同时提出一个新的加速解法，那第六句继续说我们自己干的这另外一件事情。

第七句：如果自己在方法/技术/算法/架构上做的事已经讲完了，就进入实验部分。首先写我们在什么数据集下做了一个什么规模的什么类型的实验

第八句：实验结果+对比结果。可以是量化的，在这里也可以定性的交代。

第九句：其他的实验结果和结论，比如敏感性、消融、观察到的重要现象等。看具体情况，不一定有。

3. Keywords

4~6 个

1) 尽可能不要用缩写，特别是没几个人知道的缩写，用全称。像 RNN、CNN、LSTM、SVM 是可以的，但是自己论文里提出来的缩写就不合适了。

2) 单个 keyword 不要太长，太长会显得很冗繁

3) 也不需要太多，太多也会显得太冗繁

4) 但也不要太少，1~3 个就太少了，让人觉得这篇论文的工作不重要。

5) 选的 keywords 真的就是 keywords。比如，有的同学仅是使用 Pytorch/Tensorflow/Matlab 作为求解的工具，但是却将 Pytorch/Tensorflow/Matlab 作为 keyword，就不合适了。

4. Introduction

第一段：问题的背景和来源。有不少同学第一次写论文，上来就开始写我们提出了 XXXX，这是不对的。

1) 一定要首先用一段来说明问题的来源，即研究的问题是怎么来的？来源自实际生活？还是研究本身(比如神经网络参数太庞大，神经网络可解释性不高)？还是你发现了一个新的问题？或者通过调研等发现了一个问题？问题本身的价值至少占了一个研究工作 1/3 的重要性。

2) 这个问题有什么影响。即有可能这确实是一个问题，但是如果通过一个博客或者技术报告就可以解决，而不需要一个博士生投入半年来解决，那这个问题在研究上的价值是不高了，即你的论文工作做得再好，意义也有限。所以，要向 readers 阐释这个问题的“地位”，包括这个问题造成的影响，引起的一系列问题，不解决这个问题会造成什么后果等。

3) 点出“这个问题无可避免，所以我们要解决这个问题”。

第二段：

1) 已有的工作在解决这个问题或者相关问题上做了哪些尝试和努力。如果这个问题已经有人做了，那么就总结性的写前人从哪几个角度或者采用了哪几类方法来尝试解决这个问题，需要注意：

a. 不要写成 related work。一定是总结性的写，不要一个一个地列举，这是不少同学容易出现的问题。所以可以看出，写一篇也没那么容易，列几条就可以了，处处需要总结需要归纳。

b. 一定要客观的总结，不需要写得特别表扬或者特别不表扬。

2) 如果这个问题确实没有人直接做过，也要查找一下相关问题。一般情况下，不太存在一个问题，即没有人直接做过，又没有人做过相关问题，是一个完全的孤立问题，这种情况很少。

3) 指出现有方法或者相关方法存在的问题。原因在于，如果这个问题已经被现有的方法“完美”解决了，那你这个工作的意义也有限了。一般情况下是存在一些问题的，如果实在是没有找出来明显的问题，可以从完整性上来讲，即我们做的多、考虑得全，比如之前的工作没有利用 XXX，没有发现 XXX，忽略了 XXX。

第三段：针对以上问题，in this paper，我们做了哪些贡献。这里就可以写得详细一些了。但是注意，不要和摘要，以及下面的贡献总结重复了。在第三段中，要针对“现在存在的问题，我们干了什么”，因为下面还有贡献总结，并不需要把所有的技术贡献写一遍。如果这样，那有可能出现这种情况：摘要里写了一遍，第三段写了一遍，贡献总结里又写了一遍。第三段要写的是为了解决现在的方法存在的问题，我们做了哪些努力，比如观察到了什么现象，做了什么分析，用了

什么技术手段或者基于什么模型来解决这个问题（而不需要说：我们提出一个 XXX-based XXX model，这个在贡献总结里说）。

第四段：贡献总结。一般现在的贡献总结是 3 个，如果其中一个贡献比较大，可以是 2 个。但是对于 CS 的论文，一般不会只有一个，有可能技术贡献就是一个，但是可以写上做了实验得到的观察。研究的核心在于发现（discovery），技术的核心在于发明（invention），所以在第一条写了发明，又找不出第二条发明的情况下，第二条可以写发现，包括发现的规律、现象、结果、结论，可以作为一个贡献。

贡献要写出论文的“贡献”，因为有时候同学们真正的贡献没写上，不是贡献的贡献写上了。比如有的同学写：我们做了实验。做了实验不叫贡献，通过实验得出的观察、现象、启发、发现才叫贡献，做实验本身是应该的。还有的同学写：我们用了这个方法。用了这个方法也不叫贡献，有没有新的方法提出？或者用了这个方法解决了这个问题有什么优势？别人没有发现你发现了，别人没有尝试你尝试而且有效果，这才是贡献。

如果贡献比较多，也可以是 4 个。但是 4 个就可以了，一般不会再多了，再多说明这篇论文做得比较杂，或者作者总结得不到位。

第五段： The rest of this paper is organized as follows. Section II/XXXXXXXXXXXXX.

5. Related Work

第一段：起一段，不要上来就 XXX et al. proposed XXXX [1]. 第一段以总结性的话，一般就一两句，总结归纳一下下面要说的 related work。比如下面的 related work 从三个角度来简释的，那么这一句话就总结一下这三个角度是什么。现在 CS 的论文一般情况下 related work 至少要从两个角度来说，因为从 title 里就可以看出。方法层面的 related work 写一段，问题层面的 related work 也可以写一段，环境层面的 related work 也可以写一段。

第二段：可以起一句，或者用一个名词结构加粗的形式起一个点。不要上来就 XXX et al. () developed a XXXX. 这样 readers 们进入得太快，不知道这个维度在说什么，这一段其实基本相当于没写。

第三段：同第二段。

.....

6. The Developed Framework

这一部分可以没有。一般情况是解决的这个问题不能由单一的一个目标函数或者算法解决，需要先分析特征，再设计一个正则，再 combine 起来提出一个目标函数，才能解决。甚至论文里从两个角度提出了两个模型，或者在两个领域，

或者用到了一系列特征工程，这时候可以写一个 **framework**。即告诉 **readers**：这个问题不能简单的由一个算法来解决，我们在论文里做的工作比较多。

7. The Proposed Method

论文中所提出的方法，需要注意的包括：

1) 要写出所提方法的动机，不要上来就摆公式或者画圈圈线框框图。为什么这样设计神经网络结构？为什么这样设计正则化项？为什么使用动态规划？为什么 **XXXX**？这些都是发明之前的发现，没有这些动机，只是摆公式，会让别人无从下手去理解你的工作。

2) 如果是算法，要有对应的 **line to line** 的解释，不要放一个算法在那里就完了，或者简单一句解释就结束了，要有 **line to line** 的解释。

3) 如果是目标函数，要起一个名字，作为 **model** 的名字。不要只把 **min** 或者最大似然，或者神经网络公式组一摆。

5) 不要指望 **readers** 可以通过公式来读懂你的 **model**，要通过文字的解释、图形的解释、表格的辅助、算法的辅助来全方位的阐述你的方法。世界上只有你一个人 100%懂你的论文，导师也代替不了你，更何况 **readers** 们的时间也很紧。

6) 不要将自己论文“中与不中”太依赖于自己的模型或者算法本身的叙述，不只是像神经网络、概率图模型、传统机器学习等领域，即便是在面向应用的领域，**intuition** 和 **motivation** 的重要性都比模型本身的公式要强。

7) 写完这一部分问自己一个问题：别人在只看方法没有看实验结果之前，就能够相信我的方法 **work**，能做到这一点吗？能 **convince** 别人吗？

8. Experiment and Evaluation

第一子部分：数据集。要说明数据集的统计特征，一般就是通过表格的形式给出。不要只说一个我们用了这个数据集、那个数据集。

还要说明 **training set**, **test set** 的划分方法。

第二子部分：参数设置。如果这一部分不多，可以和第一子部分合并，但是要有。

第三子部分：指标，只要不是短文，就要列出公式，理论上讲，即使最简单的 **test accuracy** 也要列出计算公式。只写一个名字肯定是不行的，至少要有一句话的解释。同时，理论上都要有计算公式。同时，简单的说明一下选择这些指标的原因，比如是在这个领域用得广泛，还是用在这里很合适之类的。

第四子部分：对比结果。有同学可能问：如果我的论文工作没有可对比的方法怎么办？那可以通过改为消融实验，自己的方法模块和自己比，也可以找基于的 **basic** 方法来比。

对比的方法一般找 **a.** 有代表性的；**b.** 比较新的。如果一个方法即没有代表

性又不新，那就没有太大必要去比。论文工作实质是一个做后感，研究性工作的核心在于发现和发明，而不是把所有方法都比下去，所以不需要担心：这么多已有工作，论文里都需要比吗？

即便有方法在某个指标上比自己的方法好，客观承认，分析原因，说明自己论文在这一问题上的发现有何不同。

这里提醒大家一句，在第四子部分的开始，有些同学习惯上来就说“为了验证我们所提出的方法的优越性/性能的优势，我们进行了对比实验。”你的方法的“优越性/性能的好”是要通过接下来的实验数据来验证的，在第四子部分的开始，实验数据还没有呢，表还在下面呢，图说不定还在下一页呢，怎么可以现在就说“我们的方法就是好”呢？

第五子部分：敏感性实验、消融实验、(error) Analysis、Case study。一定要有，如果只有对比方法，那么大家无法仅通过实验达到 convincing 的效果，然后论文大概率会被 reject。因为一个工作 solid 不 solid，仅有对比方法是远远不够的，更何况现在神经网络的调参和数据集不同带来的影响。

- 1) 模型中参数有吧，就要有敏感性实验。
- 2) 模块不同模块组成的吧，消融实验。
- 3) 模型在所有情况下都表现得好吗？Error Analysis
- 4) 结果有可展示的吗？可交互的吗？Case Study

9. Conclusion and Future Work

1) 不少同学第一次写论文 Conclusion 部分都写不对，都相当于把 Abstract 给复述一遍。Conclusion 部分是写结论，即你这篇论文的工作得到 produce 了什么，验证 verify 了什么，证实 testify 了什么，发现了什么等。重点在于这个“了”字，不是把摘要换个说法重复一遍，也不是把自己的贡献总结再写一遍。而是自己的方法得到了什么？解决了什么问题？实验结果证实了什么？证实了自己的方法有效，这个得有吧，或者验证了自己设计的算法的有效。自己在解决这个问题的过程中，观察到的现象被成功运用到了方法中，等等。重要在于“了”，在于“成功”，在于“证实”，即结论，而非预期。

2) 有不只一位同学有过这样的疑问：为什么论文的技术性贡献要写四遍？Abstract 里一遍，Introduction 第三段里一遍，Contribution Summary 一遍，Conclusion 再一遍？如果说真有逻辑上的重复性，也只有 Abstract 里与 Contribution Summary 有一定的重复性，当然，Contribution Summary 里的技术贡献会比 Abstract 里的详细。所以，这四个部分其实没有逻辑上的重复性。

3) Future work 可以不写，但其实可以写上。这个问题真的已经完全解决了吗？下一步没有想做的点了吗？一般还是有的。但不要强迫自己写，强迫写出来的 reviewer 一看就是凑来的。

10. Reference

1) 用正式发表的出处，不要用 arxiv 作为出处，arxiv 上放的论文版本理论上没有经过 peer review。

2) 实在找不到，再用 arxiv。

3) 作者、题目、标题、出处、卷期（期刊论文）、页码、年份要齐全。确实会存在一些论文没有明确的页码，这样的论文除外。书籍、网址、毕业论文等按照要求来写。