**P10**

2021-10-6 15:10:29

【问】

嗯，好，那我们开始吧。刚刚说的技术文档，你应该也听了一些，然后就大概知道是什么样的东西。我们针对的是开发人员用的文档，不是那种普通的说明书。嗯，那首先第1个就想让你简单介绍一下你的专业背景和做过的一些开发的经历吧，先不用讲的那么详细。

【答】

我本科是XX大学XX学院XX专业，然后研究生现在是XX学院的XX方向，然后现在是在XX实习，然后方向大致就是NLP方向。然后开发方面的经历，从本科的时候是有做过一些比如说网上预约这些比较小的一种网站吧，然后毕设的时候做了一些知识图谱，就是NLP的一个东西。然后研究生的时候继续做NLP的一些项目，比如做过一个基于深度学习的自动批改的系统，现在实习是在做一个，就是有一个UNIT的平台，然后上面有一些比如说关于词槽这些语义方面的分析或者是一些成分上面的分析。

【问】

你的开发的经历从本科来说大概就有四五年的时间了，从本科算起。

【答】

对，但是比较杂，刚开始是一些软件方面上的一些开发，后来可能渐渐的就是搞算法方面的。

【问】

那么就请你回忆一下你最近或你以前学过的某一个工具，某一个框架，甚至某一种编程的语言吧，然后你讲一下你是怎么学会它的，从哪听说它开始决定要学它开始，到你真正的学会它，然后用它完成一些项目，一些工作。

【答】

嗯，那就是说我最近实习的时候，因为我是在XX实习，它们是自己有一个自然语言处理的一个框架，就是那个Paddle，Paddle分为Paddle1和Paddle2这两个版本的开发。它在代码逻辑和结构上是和pytorch很像，但是是有一些不同的。然后我在平时的开发中我是参考它百度内部的一些文档吧，就是API的介绍，然后进行代码的开发，然后功能的实现。

【问】

你是怎么参考的，你就读这个文档吗？

【答】

对，就读这些文档，就比如说，因为我也不是从0到1，比如说我mentor他之前是有一些代码，就是已经完成了一些功能，然后我可能需要添加呀或者是修改，然后我就可以参照他的代码，然后比如说它用一个函数，我不知道这是干什么的，然后我就可以去那个网站搜索一下，然后搜索这个函数它具体的含义还有参数、返回的值这些东西，然后来读懂这个代码，然后再进行我自己的修改和完善。

【问】

一个就是基于你前面的人写的代码，另外一个就是基于文档，是吗？

【答】

对，或者是公司也有一个内搜，就是在上面可以看到一些公司内部其它人写的一个技术文档，就是比如说你搜一个问题，有可能他写一个博客也记录这个问题，然后对你直接点进去看是不是你想要的。

【问】

这个是在公司的内网才可以看的吗？

【答】

嗯，对是的。

【问】

哦明白，嗯，那你在遇到问题的时候，你一般会怎么做？

【答】

嗯，那遇到问题，，首先就是先自己去查嘛，自己先去看是不是函数用错了，或者这个逻辑弄错了。之后的话可以去问一下我mentor，这里到底是为什么。是环境错了还是我这个函数不对。

【问】

那你会去看文档吗？

【答】

对啊，文档是先去看的嘛。我有可能是这个语言这个框架用的不对，然后我先去看一下，先去查一下是不是我用错了。

【问】

查看文档主要为了验证它是对还是错是吧？嗯，然后文档不能解决的就问你的mentor。

【答】

对，是的。

【问】

下面就是想问一下你对于文档的认知，你觉得在开发过程中文档扮演什么样的角色？它重要吗？

【答】

嗯，比如说我刚学这个东西的时候，它就是我的一个学习资料，然后我先入门的时候对它一些基础的用法有个了解，然后在我后续开发的时候，我用到了哪个部分，然后我再去专门的去研究那一部分的一章节，然后遇到问题的话我再看一下它有没有那些，因为它那个文档，它那个网页是有那些相关问题嘛，然后它有一些问题解答，然后有可能你的问题能得到解释。

【问】

你说的网页上的相关问题就是那种Q&A是吧？

【答】

对是的。

【问】

嗯，那相当于你是分不同的阶段来用这个文档，刚开始是为了学会它，主要看它的介绍，然后深入地使用它才会去看深入的部分。

【答】

对，刚开始是一个大概的了解，然后之后的话用到哪部分我再去查看哪部分，然后遇到问题的话我再深入研究一下，然后再专门的搜索一下。

【问】

嗯那你平时都会用到哪些类型的文档？

【答】

就是这种官方的文档用的很多，嗯，然后比如说我搜索出来有一些人写的就像博客类型的那种，也是算一种。基本上就这两种。

【问】

就只有这两种，没有其它的吗？教程教材之类的会看吗？

【答】

教程我之前是会看，比如说我学这个，比如说自然语言处理它也分很多方向，比如说什么推荐方向，然后我就会看书，嗯，然后买一本书，然后来看它的各种方向各种框架的一个介绍。

【问】

那你是怎么找到这些文档？主要是通过什么样的途径来找到？

【答】

书籍的话我会看一些比如说知乎上的推荐，然后有一些公众号，我比较关注的一些资深的我觉得很靠谱的公众号，它上面有一些推荐，然后再从那些推荐里面找到我想要看的这个书。然后官方的话就是它官方给我的然后让我学的，对，然后博客就是我自己搜到的，我觉得它对我有用，然后我会继续读。

【问】

搜索引擎搜到的，官方的话就直接到官网里面去看？

【答】

是的。

【问】

那你找到这些文档之后，你是怎样阅读它们，怎么快速的阅读到自己需要的信息？

【答】

嗯，官方文档的话，我觉得它一般设计的都还是挺合理的，它会分为很多方向吧，就比如说paddle这个官方文档，它就会分为paddle1、paddle2，然后里面的话它会分为，比如说NLP方向，然后CV方向，然后NLP方向它里面会分为那种激活函数，还有一些层的结构。然后它就分的很细，只要你对NLP有了解的话，你就知道那个函数它应该属于哪部分，然后你就去那里面去找就很快，或者你不太清楚的话，直接用那个搜索，就是直接输入它的函数，然后你就能直接找到。

【问】

搜索是文档内部的搜索功能？

【答】

是的，然后书籍的话，你要学习的这个方向是一个稍微粒度大一点的一个分类，可能就到那个章节去看有没有你想要的那个问题。

【问】

通过目录章节来定位？

【答】

对，是的。

【问】

那你可以回想一下你看过的比较好的文档的例子和差的文章的例子吗？可以大概说说它们的优点缺点。

【答】

嗯，好的文档我认为就写的比较详细明白吧，而且它就不会误导你。

【问】

可以举个例子嘛？

【答】

好的文档的例子，因为我之前用c++开发比较少，那我现在是需要用c++开发，所以我也有用c++的官方的一个技术文档。它都是英文的，但是我觉得它英文对于每个方法它都写得比较简单，就是很直接，然后下面也会给一些例子，然后看起来就，有的时候可能你觉得看英文看起来字很多，或者你直接看它那个例子，然后你就明白它这个函数是在干什么了，为什么得到这个结果，我觉得这样就挺好的。

【问】

它给的例子比较好，能够让你比较直观的学会它是吗？

【答】

是的，我觉得这个例子可能比它的语言的介绍应该更直接一点。

【问】

例子是有带有代码的那种例子吗？

【答】

对它有代码，然后给你一个输入输出，而且它有好几种，就是它有一些对比，就是函数之间的对比，就比如说有的函数返回的是经过某种处理的，另一个函数经过不同的处理，就能立马看出来。

【问】

除了这个例子比较好，那还有其它的优点吗？

【答】

嗯，其它优点，因为官方的肯定是比较准确的，如果你去网络上直接搜索的话，会出现很多结果，比如说你在CSDN上面得到的博客，它那个质量参差不齐，就觉得它说的是云里雾里的，或者是说得不对。官方的就应该比较权威一点。

【问】

语言上呢，语言怎么样？

【答】

语言就很直接，很简单易懂。

【问】

结构呢？

【答】

结构的话，因为一般我去读文档的时候，我是遇到一些问题，然后我就去直接搜索，一般就能很快得到想要的结果，我不会说从头到尾的看对，所以也不太关注这个结构。

【问】

明白了，哦，那你可以举一些比较差的例子吗？

【答】

差的例子，因为技术博客也算一种技术文档，网络上这种博客可能是越来越多，然后水平有限。比如说那个CSDN，我之前用的时候，我感觉它的水平还可以，我觉得它那些教程还可以的，还挺准确的，但是后来我感觉它就很乱，就很难找到一篇正确的，很有价值的博文。

【问】

除了CSDN，你还用过国外的这种论坛吗？

【答】

国外的，就是那个stack overflow，那个挺好的，就我看它提一个问题，然后后面有很多回答，然后一般排到最前面的那个回答就是最好的吧，就是最正确的那个回答。然后就非常方便，解决问题的时候非常高效。

【问】

那除了这种非官方文档，官方文档里面你见过比较差的吗？

【答】

官方文档，我感觉我没用过什么太差的，因为我用之前我也会就是找一下是不是有什么比较好的，然后我就直接用那个比较好的了，比较差的一般也不会用。我觉得比较差的那种官方文档也没什么人用，就大家都觉得不好用，也没什么人用，然后渐渐就消失了。

【问】

下一个问题是关于文档反馈的，你见过那种又反馈的文档吗？

【答】

就是我提了一个问题，它给我解答是吗？

【问】

就是在文档的底部或者某个地方，给你留一个反馈的途径，比如说你可以提意见，提问题，或者是点个赞，或者是给你留个联系方式啥的。

【答】

嗯嗯，这种应该还挺多的，比如说，因为我是在公司里面，我mentor让我加那个paddle的解答的群，在里面就可以直接问他，然后就会有相关的人值班，然后会非常及时地回答你。

【问】

除了你们公司自己的，你用过的其他的文档也有反馈的途径吗？

【答】

一般你可以在下面直接评论吧，然后他看到的话就会回答你。

【问】

那你会评论吗？

【答】

我有，比如说那个教程写得挺详细的挺好的，但是我可能某一步跟它有一些不一样，然后我就会给它提一些我的意见，或者问它这样是为什么。

【问】

嗯，那是什么因素来促使你评论呢？

【答】

就是完善一下吧，可能这个文章我觉得可能很多人会看，然后我遇到了一个其他人没有遇到的问题，那我提的这个意见对他也是有好处的，可能别人遇到跟我相同的问题也能得到一个解答吧。

【问】

下面一个问题是，如果有一个比较理想化的状态能够让你得到一个非常好的文档，你期待这份文档是什么样子的？你可以总结几个点，你期待的好的文档的特点。

【答】

我觉得它可以针对不同程度的人写一个不同的文档，比如说有初学者，就要手把手一步一步的，然后对于那种已经入了门，有一定基础的，就应该有目的性地，就比较简单的地方可以写得简略一点，就有详略。

【问】

你是说这个文档要详略得当，该简单的地方简单该详细的地方详细吗？

【答】

对，它是有重点的。

【问】

一个是有重点，还有你是不是其他分为两个不同的版本？

【答】

可以，对，我觉得可以。因为你看博客的时候，有的人他可能水平高一点，然后他说的就比较简略。然后有的人可能就小白一点，他写得就详细一点，然后就有时候觉得有点啰嗦，反正就是有不同的感受。

【问】

那除了这个之外还有吗？

【答】

还有就是保持一定的实时性吧，因为我觉得，比如我现在用的paddle这个东西也是在不断地迭代的，他也有一些版本之前不同，我希望技术文档也更新的快一点，然后就能跟上这个迭代的速度，然后就是不会出现正在使用它的人出现什么问题，然后查文档的时候可能查不到的情况。

【问】

还有吗？

【答】

还有就是看一些国外的技术文档的时候，我是习惯是看英文，就是看它原版，但是我之前的时候也看过一些中文的，就是翻译过来的，就是希望它能够翻译的不要那么死板，有的时候就是单纯的从它那个表面意义上来翻译，但是我觉得那个翻译过来其实就跟它的实际含义不符，就是你搞不懂有什么联系。

【问】

嗯，那像你读过的这种翻译的版本里面，像一些英文里面特有的，比如说变量它也会翻译吗？还是说它保留那个变量的名字？因为代码都是英文的，有的人就反馈说还不如直接看英文的。翻译的时候会直接保留英文的关键词吗？

【答】

对，它会保留。就是那种中英文混杂是吗？中英文混杂我也能接受，因为确实有一些名词，比如handle，它有时候翻译成句柄之类的，然后我就不知道它是什么意思。你就直接保留handle，翻译不好的就保留吧。

【问】

还有要补充的吗？你觉得文档比较好的点。

【答】

应该也没有什么了吧，反正还是确实就是看一些英文的原版的就是更舒服一点，但是就比如说我刚开始学那个深度学习什么的，一般都会看那个统计学习方法那些书，然后它实际上也是从国外的翻译过来的，到那时我觉得它翻译的还好，就是理解上，那些词，，那些知识的转化，就还是挺接地气吧，就是我能理解得比较明白。像学习资料就看一些中文的还可以，你要是直接看代码，就是英文的比较好，就是具体到某个函数，那些看英文的比较好。

【问】

除了语言，还有其他地方要补充吗？

【答】

应该没有了。

【问】

最后问一个关于程序员的问题。你觉得在中国，程序员和其它专业的人比起来有什么不同？

【答】

我觉得在公司里程序员之间比较平等吧，就是虽然有经历、mentor、leader这种，但是就像个师兄，很平易近人，然后你有什么问题就直接问就好了，有什么疑问就直接问。这样也能提高你的工作效率，沟通起来也很快。我觉得这一点应该是不同。

【问】

是不是因为你见到的程序员都比较年轻，就年龄差距不大？

【答】

确实，部门里面我感觉大部分都是20几岁，可能leader那一层可能30来岁左右，对，经理可能40岁左右，但是我觉得就是经理、leader它们心态上也挺年轻的，就是没有一副特别那种成年人那种气质，就还好。

【问】

然后从学习，从生活上来看呢？

【答】

从学习生活上来看，我觉得就你学习上有什么不理解的地方，它们也能很快的帮助你吧，不会不耐烦或者怎么的，我觉得都挺好的。

【问】

那你自身的一个学习习惯是什么样的？

【答】

自身的话，因为你做一个项目你可能只需要了解这一方面的知识，但是如果你换一个方向，你换一个项目，换一个活来做的话可能又是另一方面的知识，所以你不断地干活，不断地学习新的东西，不断的了解。

【问】

那有这么一个持续学习的需要，那更多的还是要自己自学吗？

【答】

对，更多是自学，大部分是需要自己学习的，但是这个项目肯定有一些人早就接触了，你作为一个新加入的，刚开始做，你自己先学习，你有疑问可以问一些资深的人，然后就是一边沟通一边干活这么一个过程。

【问】

还是倾向于自学，应该大部分时间都是自学。

【答】

对，就是自己看一些代码，查一些资料，遇到问题也是自己先解决，解决不了的再去问。

【问】

其他的没有了，今天的访谈就先到这。