Tower of Babel in Fanworks: An Analysis of AO3 Tags / 同人创作中的巴别塔：AO3标签分析

Yunyan Duan

2020-01-06

写在前面

本文又名《一个丧心病狂的Python画图指南》，创作动机是放了个冬假于是想克服一下对用Python画图的恐惧，做完之后怨念并没有消除（没有比ggplot更好用的画图模块了（这话好耳熟哦上次从Matlab换到R的时候也哭嚎了好一阵子“还是Matlab画图好看”呢 ^_^ 有很多画图的功能是为了满足我个人的审美而搞得复杂，绝非必要。

除了这个动机以外，选择做一下AO3是出于个人爱好，而非出于专业考量，即，作者不以同人创作为研究方向，作者没有社会学/传播学/文学比较背景，大概率有比AO3更适合用来做同人创作分析的对象，等等。题目中的巴别塔，所观察到的跨语言的差异，很大原因是平台本身的特性决定的（谁是用户、平台在哪里，等等），往好了说是管中窥豹，往坏了说是以偏概全。因此，以下大量探索性分析，大量看图说话，没有验证性分析，没有理论，没有假设检验。请不要把它当做严谨的分析，请不要过度解读。

欢迎查看/玩代码，也许你能用它回答你所好奇的问题。时间精力所限，代码有bug（嗯，我知道至少一个地方有bug，下文有标注），封装和组织逻辑有不少欠考虑的地方。我所设想的读者是对同人创作有好奇心，能看懂我下文中信息量巨大的图，也有能力自行查阅、修正代码的朋友。目前我没有维护代码的打算，更多将此视为一个存档/一个用Python画图的参考。如果你发现bug，请自行修正。

以上。

1. 背景介绍

Archive of Our Own (https://archiveofourown.org/) （AO3）是一个同人创作平台，供粉丝上传、分享同人小说等同人相关的创作。它区分出10类粉圈（fandom），大致是按传媒载体（Media）做区分：动漫（Anime & Manga），书籍与文学（Books & Literature），卡通漫画与图画小说（Cartoons & Comics & Graphic Novels），电影（Movies），音乐与乐队（Music & Bands），戏剧（Theater），电视（TV Shows），电子游戏（Video Games），名人与真实人物（Celebrities & Real People），以及其他传媒（Other Media）。根据AO3自己提供的统计，它有逾3万5千个粉圈，200多万用户，500多万篇同人创作，可见是一个规模颇为可观的平台。

通过对这些粉圈的分析，我们可以对以下问题做一些初步的回答：

- 同人创作的热情是否受到粉圈的传媒载体的影响?
- 同人创作的热情是否受到原作的语言文化的影响?
- 哪些粉圈”产粮“最多?
- 在各种语言、各种传媒载体中，粉圈有哪些跨语言/传媒的共性和特性?

也许看到这些问题的时候你心里已有一些答案，不妨与下文的图表相互印证，会是颇为有趣的经历。

2. 数据

各个传媒的粉圈数据由一个简单的爬虫从AO3获取，数据截止2019年12月22日。不用爬虫也很容易获得，打开页面复制粘贴即可。对每个粉圈，使用TextBlob (https://textblob.readthedocs.io/en/dev/api_reference.html)的 `detect_language()` 来标注它的语言。粉圈通常包含原语言作品/作者以及相应的英文翻译，因此仅针对原语言部分来标注。标注的语言代码参见此页 (https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes)。需要注意的是所标注的语言并非完全准确，比如说Harry Potter -

J.K. Rowling被标成了da（Danish，丹麦语）。另外，也有一些粉圈使用了其他语言作为名字，但仍应被考虑为其原语言的原作，比如One Piece（《海贼王》）应被视为ja（Japanese，日语）作品而非en（English，英语）作品。因此对一些知名度极高的作品，作者手动做了一些更正，可参见代码。

对数据有以下*未必真实的*假设：粉圈名字无重复，同人创作对粉圈的标注准确无误，不考虑粉圈之间的层级关系。其中不考虑粉圈之间的层级关系几乎必然是错误的，比如Batman是一个粉圈，但在计算DC粉圈时也会把Batman算上，因此DC粉圈会有远比Batman粉圈更高的产粮量。出于这个原因，以下的很多分析都基于单纯的粉圈计数，而非产粮量加和。

可调用以下函数获取数据（注：ao3_functions.AnnotateLanguage()跑起来很慢，请谨慎评估时间以选择大小合适的样本量）：

In [ ]:
```
import ao3_functions

all_media = ['Anime & Manga', 'Books & Literature', 'Cartoons & Comics & Graphic
 Novels',
          'Movies', 'Music & Bands', 'Other Media', 'Celebrities & Real People',
          'Theater', 'TV Shows', 'Video Games']

mydf = ao3_functions.CrawlTags(all_media)
# -- optional (for the sake of time to run AnnotateLanguage):
mydf2 = mydf.sample(n = 100)
# -- end optional
mydf3 = ao3_functions.AnnotateLanguage(mydf2)
mydf4 = ao3_functions.CorrectLanguage(mydf3)
```

数据如下表。几个重要的column：

- fantom: 粉圈，通常是原作的名字，可能还有作者、年份、媒体的信息
- cnt: 产粮量，这个粉圈有多少作品
- MediaType: 传媒载体
- org_lang: 原作的语言
- isEn: 原作是否为英文

In [2]:
```
print(mydf4.shape)
mydf4.sample(n = 5)
```

```
(33612, 9)
```

Out[2]:

	fantom	cnt	href	MediaType	index	org_ftm	
700	Free!	16317	/tags/Free!/works	Anime & Manga	700	Free!	
20107	Lighthouse - The Hush Sound (Song)	1	/tags/Lighthouse%20-%20The%20Hush%20Sound%20(S...	Music & Bands	1879	Lighthouse - The Hush Sound (Song)	Lig - T
23982	H2O: Just Add Water RPF	1	/tags/H2O:%20Just%20Add%20Water%20RPF/works	Celebrities & Real People	534	H2O: Just Add Water RPF	H Ad
13601	Felicity - An American Girl Adventure (2005)	6	/tags/Felicity%20-%20An%20American%20Girl%20Ad...	Movies	2046	Felicity - An American Girl Adventure (2005)	A A
2537	Ultra Battle Satellite (Manga)	1	/tags/Ultra%20Battle%20Satellite%20(Manga)/works	Anime & Manga	2537	Ultra Battle Satellite (Manga)	

然后是准备调色盘（你也大可跳过这一步骤），以保证各传媒、语言各自对应于~~符合我审美的~~固定的颜色，以及一些语言的stop words：

In [3]:
```
mymclr, mylangclr = ao3_functions.get_my_palette()
mystop_lang_dict = ao3_functions.get_stop_words()
```

数据准备就绪，下面就可以开始愉快玩耍了。

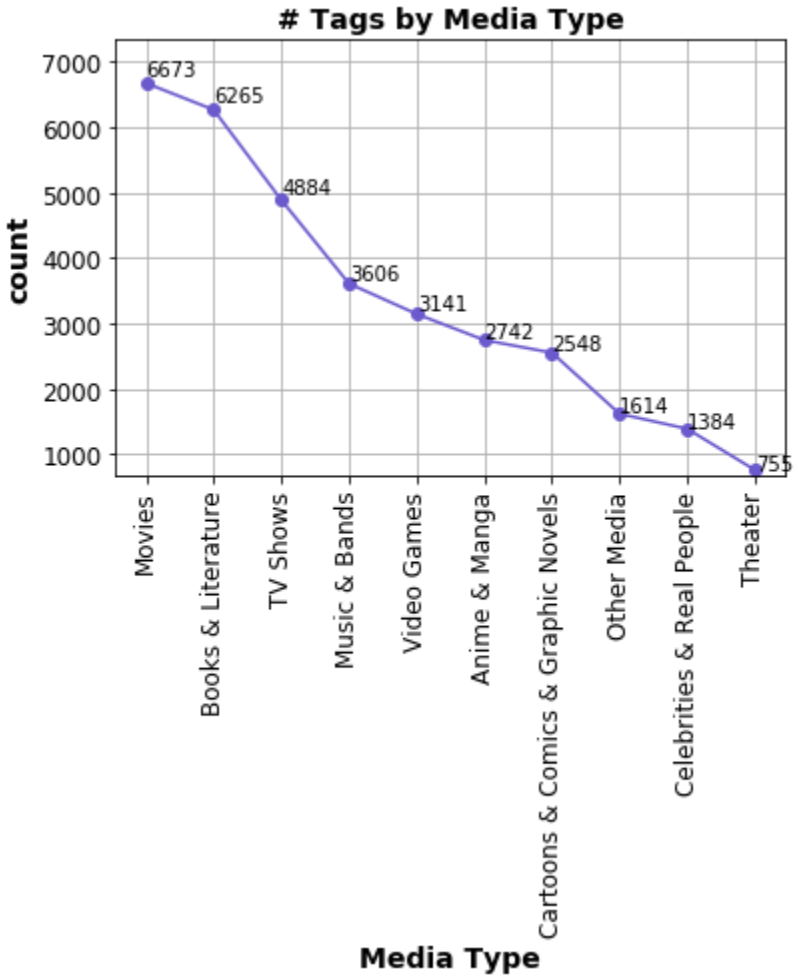
3. 分析

3.1. 各类传媒各有多少粉圈?

以粉圈的数量来论传媒的英雄或许不够公正，但它能在某种程度上反映同人创作受传媒载体的影响。换言之，某些传媒可能比其他传媒更能激发粉丝的创作热情，也就产生更多的粉圈。

如下图可见，电影和书籍产生了数量最多的粉圈，都在6000+，最少的是戏剧，只有755。这大约是因为戏剧的传播并不如电影和书籍那样广泛，戏剧作品数量本身也不多。

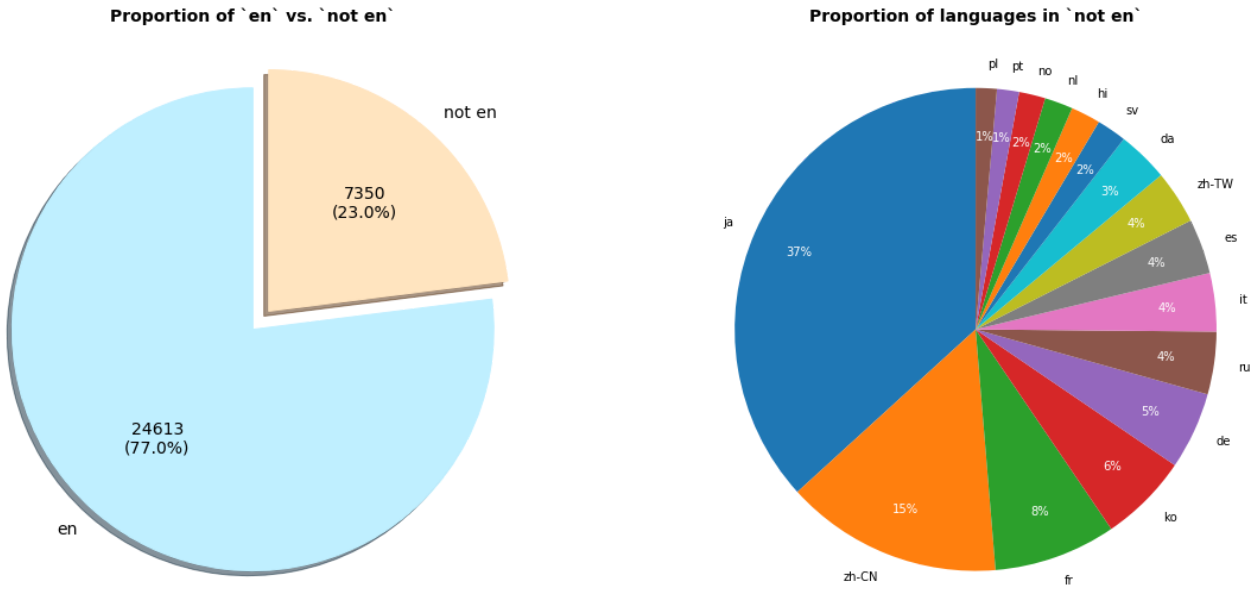
In [4]:
```
dfm, fig = ao3_functions.Vis_TagCntPerMedia(mydf4)
```



3.2. 各种语言各有多少粉圈?

Again，以粉圈的数量来讨论语言的影响力不够公正，但它仍然有不少信息。鉴于大约有3/4的粉圈都是英文，其他的语言被单独拿出来看了一下比例。日语（ja）名列第一，占比37%，简体中文（zh-CN）紧随其后，占比15%，二者合计已占据半壁江山。接下来法语（fr）、韩语（ko）、德语（de）、俄语（ru）、意大利语（it）、西班牙语（es）、繁体中文（zh-TW）和丹麦语（da）分列3-10位，占比都在3%及以上。（语言的对应颜色也可以改，但没精力改了，就这样吧，瘫

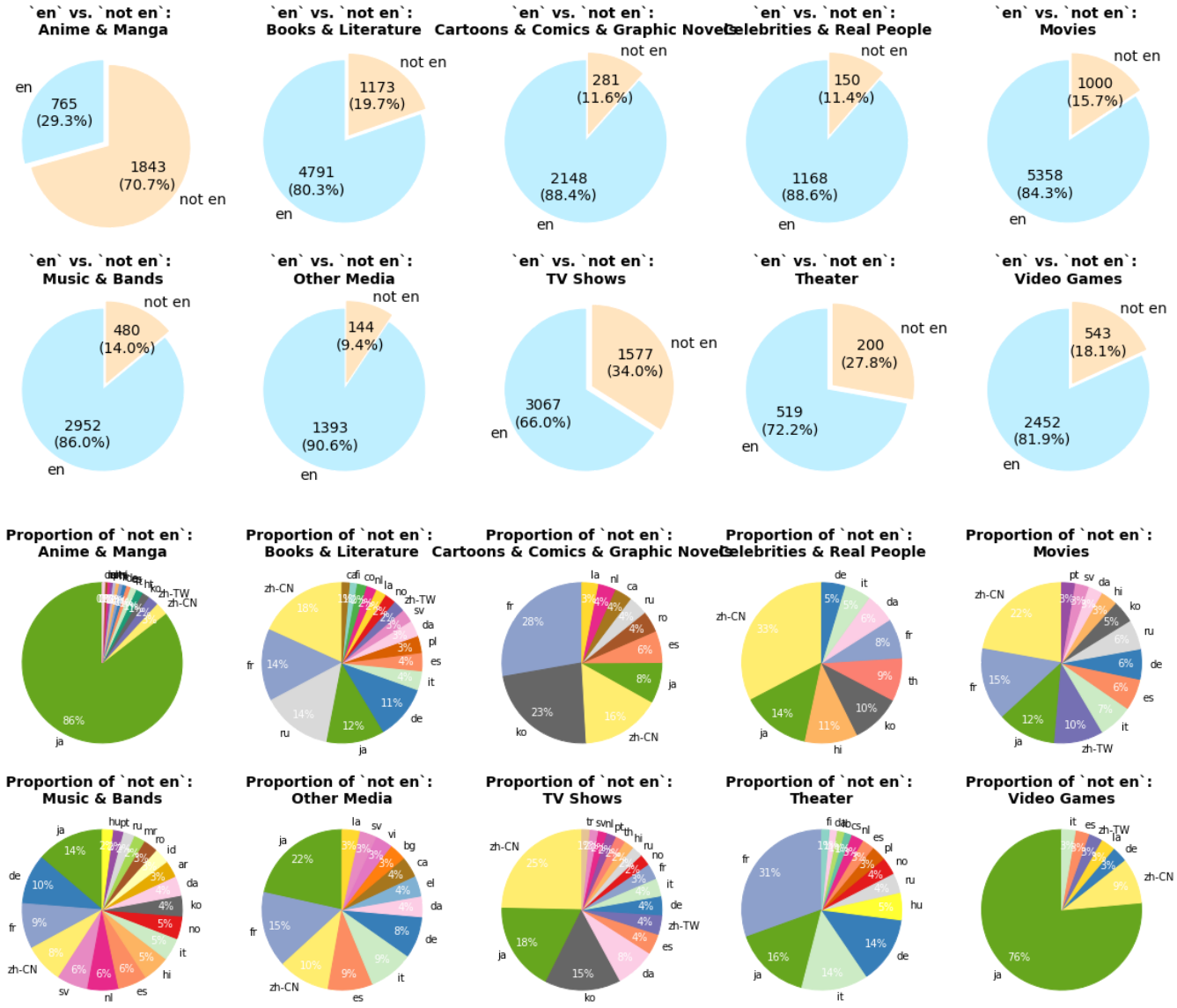
In [5]:
```
dfl, fig2 = ao3_functions.Vis_TagCntPerLang(mydf4, min_cnt = 100)
```



3.3. 每种媒体形式里，各种语言的影响（粉圈数量）相同否?

这大概是一个巴别塔所在之处：在不同的媒体形式中，不同语言的粉圈占比并不相同。在除动漫以外的传媒形式中，英语都占到了绝大多数，又以真人（88.7%）和卡通漫画（88.4%）为最。在动漫中，日语占绝大多数（85%），在电子游戏中也是类似的模式。书籍与电影的模式也颇为近似，简体中文和法语分列前二位。卡通漫画占据前两名的是法语和韩语。真人和电视的模式类似，中文、日语、韩语名列前茅。戏剧中比重最大的是法语、日语、意大利语。音乐是最平均的一类，每种语言占比差距不大。（的确，音乐不怎么受语言影响，基于音乐的同人创作看起来也是如此~

In [7]:
```
# 语言太多了，只取每一类媒体中占比由高到低排列，直至加和占95%的语言
d,f1,f2 = ao3_functions.Vis_LangByMedia(mydf4, ncol = 5, quantile = 0.95, lang_color = mylangclr)
```

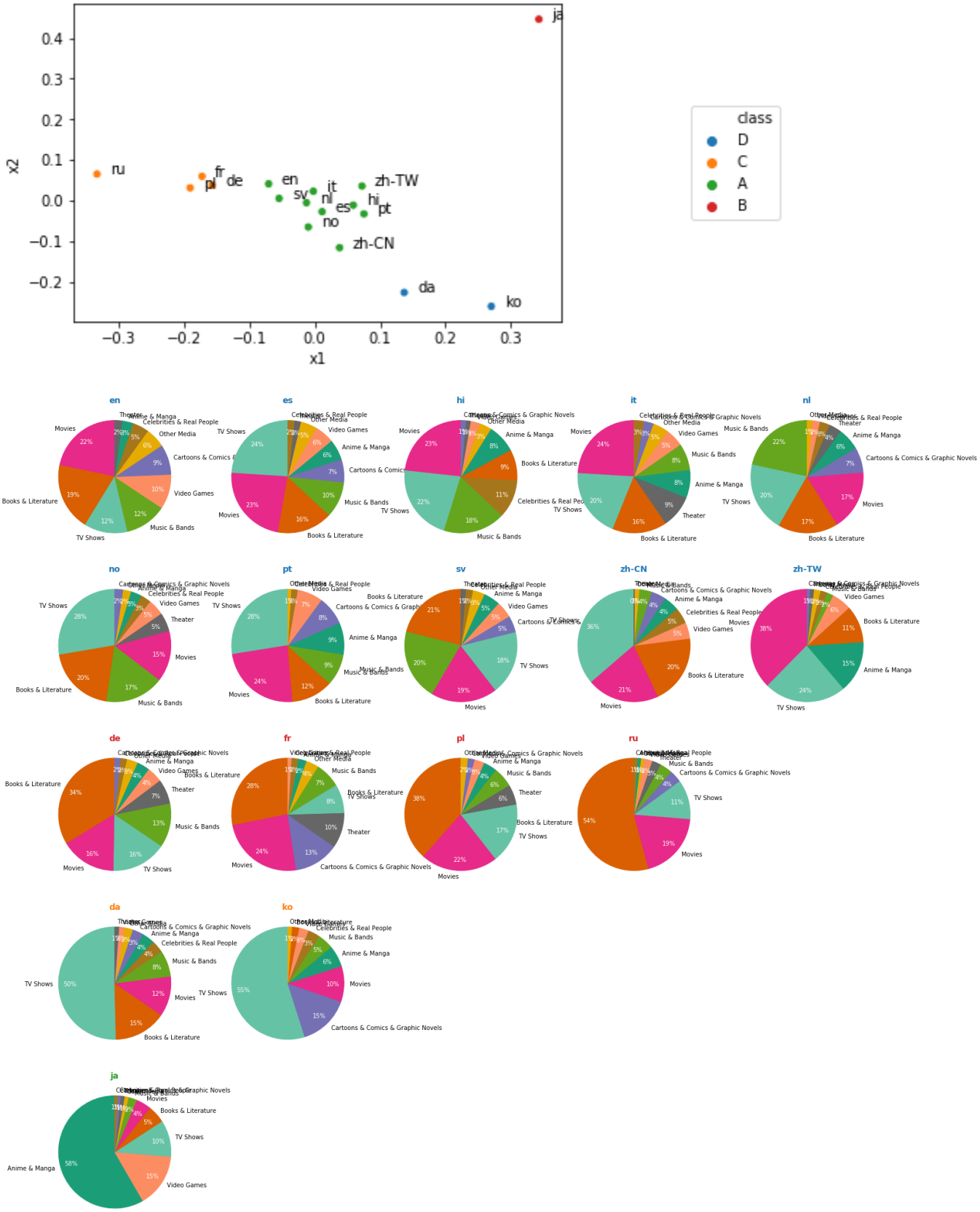


3.4. 每种语言里，各种媒体对粉圈数量的影响如何?

在这里做了一个~~并无卵用的~~尝试，想看看能否根据各语言在各类媒体的粉圈数量分布找出相似的语言和不相似的语言，所以做了个降维和聚类。降维用的PCA，两个成分能解释75%左右的变异。数据量比较小，~~无视outlier~~强行KMeans分了4类：

- 第一类：en, es, hi, it, nl, no, pt, sv, zh-CN, zh-TW （并看不出有什么pattern，从PCA结果图来看是把outlier分出去之后剩下的大类）
- 第二类：de, fr, pl, ru （很一致的，注重书本、电影、电视的类别，戏剧占比也不小）
- 第三类：da, ko （电视占比高于50%，ko我懂，但da我不懂……难道是数据量太小标准化完了就偏了吗）
- 第四类：ja （一枝独秀的动漫和电子游戏）

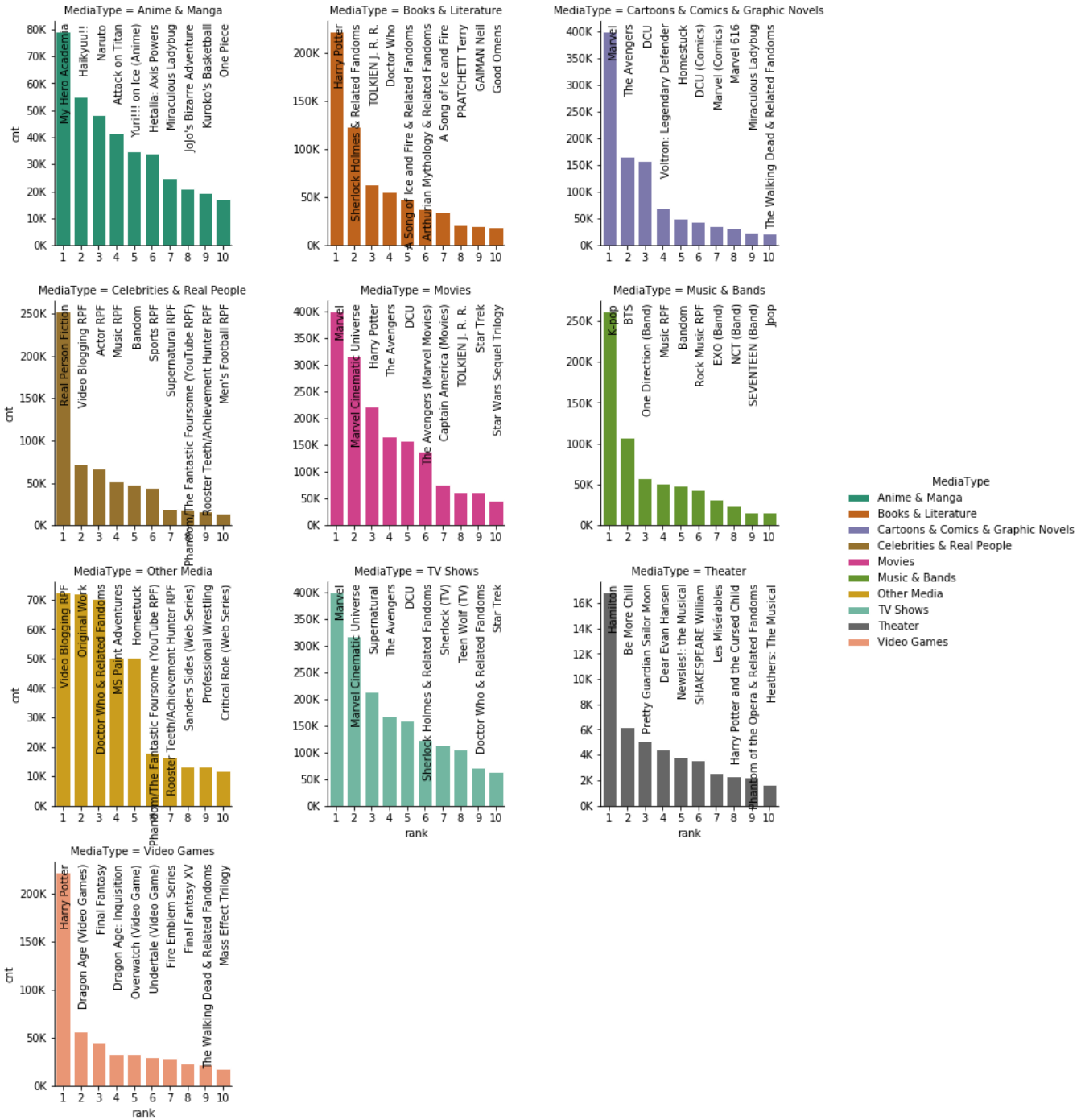
In [8]:
```
tmplangs = list(dfl['org_lang'])
tmp, fig, figclst = ao3_functions.Vis_MediaByLang(mydf4, tmplangs,
                    do_cluster = True,
                    max_ncol = 5,
                    ncluster = 4,
                    media_color = mymclr)
```



3.5. 每种媒体中产粮最多的粉圈有哪些?

按传媒形式和产粮量排了个序，显示前10名的产粮量与英文名~~（因为写不出显示其他语言的代码子）~~。大致的分布还挺Zipf的。然后欢迎在这凌乱的图中寻找熟悉的/预料中的结果。比方说，在书本与文学类里，排名第一是哈利波特，产粮量在200K+；排名第二是福尔摩斯及相关粉圈，产粮量100K+；第三是托尔金的小说，包括大名鼎鼎的《魔戒》《霍比特人》等等，产粮量50K+。再往下是神秘博士、冰与火之歌、亚瑟王传说，产粮量50K左右。看看这些作品的名字翻译成中文毫不费力，可见其影响力很大。

In [10]:
```
tmpa, tmpb = ao3_functions.Vis_PopularTags(mydf4, top_k = 10, media_color = mymclr)
```



~~因为图画不出来英语以外的其他语言所以~~在表里看一下中文的产粮能力。有没有看到眼熟的名字？当然也有一些明显的错误（动漫这个类别比较突出）。还有就是分类模糊的问题了，”声入人心“横跨三类（真人、音乐、电视），高居榜首。（这个结果真的很make sense……

In [11]:
```
ao3_functions.PopularTagsPerLang(mydf4, ['zh-CN'], top_k = 5)
```

Out[11]:

MediaType	Anime & Manga	Books & Literature	Cartoons & Comics & Graphic Novels	Celebrities & Real People	Movies	Music & Bands	Other Media	TV Shows	Theater	V
rank										
1	一人之下 The Outcast | Hitori no Shita: The Outcas...	魔道祖师 - 墨香铜臭 | Módào Zǔshī - Mòxiāng Tóngxiù	凹凸世界 | AOTU Shijie | AOTU World	声入人心 | Super-Vocal (TV)	哪吒之魔童降世 | Né Zhā Zhī Mó Tóng Jiàng Shì (2019)	声入人心 | Super-Vocal (TV)	HIStory3 - 圈套 | HIStory3: Trapped	声入人心 | Super-Vocal (TV)	全职高手舞台剧 - 朱璐莎 | The King's Avatar Stage Play -...	I
2	罗小黑战记 | Luó Xiǎo Hēi Zhàn Jì | The Legend of L...	全职高手 - 蝴蝶蓝 | Quánzhí Gāoshǒu - Húdié Lán	全职高手 | The King's Avatar (Cartoon)	偶像练习生 | Idol Producer (TV)	流浪地球 | The Wandering Earth (2019)	偶像练习生 | Idol Producer (TV)	HIStory2-越界 | HIStory2: Boundary Crossing (Web...	偶像练习生 | Idol Producer (TV)	0	I
3	天狼 Sirius the Jaeger | Sirius the Jaeger (Anime)	人渣反派自救系统 - 墨香铜臭 | The Scum Villain's Self-Savi...	魔道祖师 | Módào Zǔshī (Cartoon)	陈情令 | The Untamed (TV) RPF	盗墓笔记 | Time Raiders (2016)	创造101 | Produce 101 (China TV)	HIStory3: 那一天 | HIStory3: Make Our Days Count	镇魂 | Guardian (TV)	0	
4	灵契 | Ling Qi | Spiritpact	天官赐福 - 墨香铜臭 | Tiān Guān Cì Fú - Mòxiāng Tóngxiù	19天 - Old先 | 19 Days - Old Xian	创造101 | Produce 101 (China TV)	红海行动 | Operation Red Sea (2018)	乐华七子NEXT | NEX7	上瘾 | Addicted | Heroin (Web Series)	陈情令 | The Untamed (TV)	0	'
5	GaoGaiGar	盗墓筆記 - 南派三叔 | The Grave Robbers' Chronicles - ...	秦时明月 | Qín Shí Míngyuè | The Legend of Qin - A...	镇魂 | Guardian (TV) RPF	骗爱天团 | The Fraud Love Group (2016)	中国新说唱 | The Rap of China (TV)	男生学院自习室2 | Study Room of Boy's School 2	琅琊榜 | Nirvana in Fire (TV)	0	

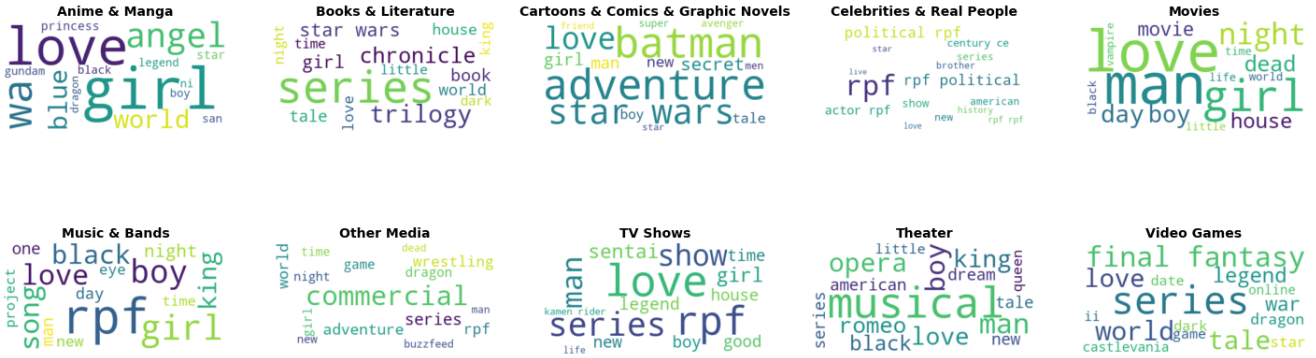
3.6. 词云1：各类媒体下的粉圈关键词

虽然各种语言文化在同人创作领域似乎存在着不同的偏好，要想更进一步地了解这是否在标签中有所体现及其具体的含义，我们仍然需要一致的语言。故此，以下展示粉圈的英文翻译的词云（没有很认真地做tokenization，随便用空格分的，简单去除英语及对应原语言中的stop words）。

纵观所有媒体，series十分醒目，在书本、电视和电子游戏里都很显眼，可见搞一个系列的作品对同人创作的启发力度很强。猜想能搞一个系列意味着能搞一个完整的世界观，所以也能吸引同人创作吧。然后love, girl, boy, man也很醒目（为什么没有woman……），大概爱情故事和”意难平“也是同人创作的动力了，在动漫、电影、电视里常常出现。音乐和真人则是rpf傻傻分不清楚。电子游戏、卡通漫画和书本里，adventure, legend, chronicle, tale十分醒目，这大约也反映了对传说和冒险的向往。

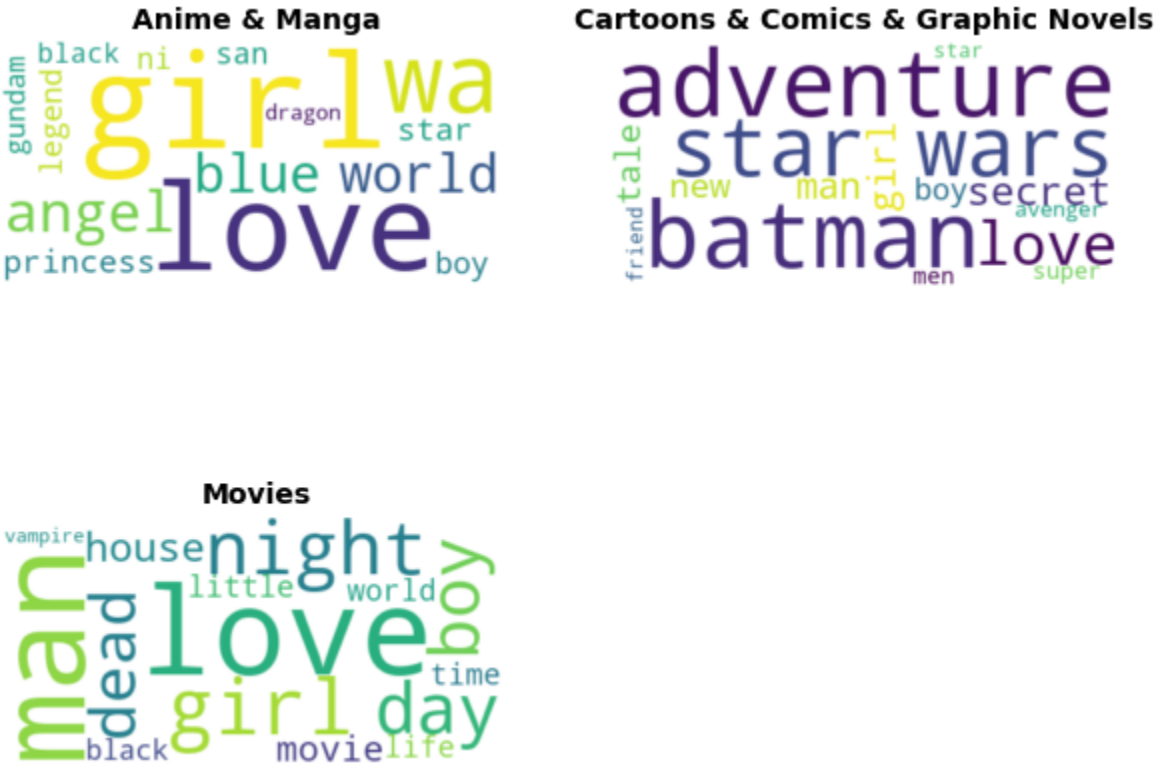
In [12]:
```
mydf5 = ao3_functions.PrepEngText(mydf4, mystop_lang_dict)
tmpmedias = list(set(list(mydf5['MediaType'])))
tmpmedias.sort()

word_cloud_all = ao3_functions.Vis_WordCloud(mydf5, {'MediaType': tmpmedias}, nco
l = 5)
```



然后重点看一下我一度傻傻分不清楚的动漫vs.卡通漫画（这里有个bug，画图的时候如果小于等于2张图会报错，所以拉上电影来凑数）。题目的区别挺明显的（again，题目的区别不完全反映题材的区别）。直观地看，动漫与电影的相似度高于与卡通的相似度。

In [13]:
```
word_cloud_ac = ao3_functions.Vis_WordCloud(mydf5, {'MediaType': ['Anime & Manga'
, 'Cartoons & Comics & Graphic Novels',
          'Movies']}, ncol = 2)
```



3.7. 词云2：各语言中的粉圈关键词

除了醒目的series, rpf之外，还是可以看出一些具有文化特色的东西，比如法语（fr）的paris，波兰语（pl）的witcher，俄语（ru）的adventure，西班牙语的zorro，繁体中文（zh-TW）的dragon。

In [14]:
```
word_cloud_by_lang = ao3_functions.Vis_WordCloud(mydf5,
                                                 {'org_lang': tmplangs},
                                                 max_word_per_cloud = 10, ncol =
5)
```

4. 结语

各种看图说话的分析已经在前一个部分写完了。如果你对粉圈没有那么了解，那么这是一个从传媒与语言角度对粉圈的概述。如果要扣题，那么说在AO3的粉圈标签里，各语言文化在它们的同人偏好上是明显有别的。然而受限于较小的数据量，短小的粉圈标签，以及极偏的英语标签数量远大于其他语言的分布，很难说有什么有效的结论。过度相信这些数据可能反而会强化对各语言文化的刻板印象，这似乎是宏观而粗浅的分析所难以回避的。除此以外，它的贡献在于如何用Python画图，尤其是如何在一张大图里组织subplots。

致谢

感谢@asherpen向我介绍manga与comics的区别，感谢@xshenshou对早期结果的建议。与你们的交流是我坚持写到这里的动力。