互联网时代的社会语言学：基于SNS的文本数据挖掘

今年上半年，我在人人网实习了一段时间，期间得到了很多宝贵的数据，并做了一些还算有意义的事情，在这里和大家一块儿分享。感谢人人网提供的数据与工作环境，感谢赵继承博士、詹卫东老师的支持和建议。在这项工作中，我得到了很多与众人交流的机会，特别感谢 OpenParty 、 TEDxBeijing 提供的平台。本文已发表在了《程序员》杂志，分上下两部分刊于 2012 年 7 月刊和 8 月刊，在此感谢卢鹄翔编辑的辛勤工作。由于众所周知的原因，《程序员》刊出的文章被和谐过（看到后面大家就自动地知道被和谐的内容是什么了），因而我决定把完整版发在 Blog 上，同时与更多的人一同分享。对此感兴趣的朋友可以给我发邮件继续交流。好了，开始说正文吧。

作为中文系应用语言学专业的学生以及一名数学 Geek ，我非常热衷于用计算的方法去分析汉语资料。汉语是一种独特而神奇的语言。对汉语资料进行自然语言处理时，我们会遇到很多其他语言不会有的困难，比如分词——汉语的词与词之间没有空格，那计算机怎么才知道，“已结婚的和尚未结婚的青年都要实行计划生育”究竟说的是“已 / 结婚 / 的 / 和 / 尚未 / 结婚 / 的 / 青年”，还是“已 / 结婚 / 的 / 和尚 / 未 / 结婚 / 的 / 青年”呢？这就是所谓的分词歧义难题。不过，现在很多语言模型已经能比较漂亮地解决这一问题了。但在中文分词领域里，还有一个比分词歧义更令人头疼的东西——未登录词。中文没有首字母大写，专名号也被取消了，这叫计算机如何辨认人名地名之类的东西？更惨的则是机构名、品牌名、专业名词、缩略语、网络新词等等，它们的产生机制似乎完全无规律可寻。最近十年来，中文分词领域都在集中攻克这一难关。自动发现新词成为了关键的环节。

挖掘新词的传统方法是，先对文本进行分词，然后猜测未能成功匹配的剩余片段就是新词。这似乎陷入了一个怪圈：分词的准确性本身就依赖于词库的完整性，如果词库中根本没有新词，我们又怎么能信任分词结果呢？此时，一种大胆的想法是，首先不依赖于任何已有的词库，仅仅根据词的共同特征，将一段大规模语料中可能成词的文本片段全部提取出来，不管它是新词还是旧词。然后，再把所有抽出来的词和已有词库进行比较，不就能找出新词了吗？有了抽词算法后，我们还能以词为单位做更多有趣的数据挖掘工作。这里，我所选用的语料是人人网 2011 年 12 月前半个月部分用户的状态。非常感谢人人网提供这份极具价值的网络语料。

要想从一段文本中抽出词来，我们的第一个问题就是，怎样的文本片段才算一个词？大家想到的第一个标准或许是，看这个文本片段出现的次数是否足够多。我们可以把所有出现频数超过某个阈值的片段提取出来，作为该语料中的词汇输出。不过，光是出现频数高还不够，一个经常出现的文本片段有可能不是一个词，而是多个词构成的词组。在人人网用户状态中，“的电影”出现了 389 次，“电影院”只出现了 175 次，然而我们却更倾向于把“电影院”当作一个词，因为直觉上看，“电影”和“院”凝固得更紧一些。

为了证明“电影院”一词的内部凝固程度确实很高，我们可以计算一下，如果“电影”和“院”真的是各自独立地在文本中随机出现，它俩正好拼到一起的概率会有多小。在整个 2400 万字的数据中，“电影”一共出现了 2774 次，出现的概率约为 0.000113 。“院”字则出现了 4797 次，出现的概率约为 0.0001969 。如果两者之间真的毫无关系，它们恰好拼在了一起的概率就应该是 0.000113 × 0.0001969 ，约为 2.223×10^{-8} 次方。但事实上，“电影院”在语料中一共出现了 175 次，出现概率约为 7.183×10^{-6} 次方，是预测值的 300 多倍。类似地，统计可得“的”字的出现概率约为 0.0166 ，因而“的”和“电影”随机组合到了一起的理论概率值为 0.0166 × 0.000113 ，约为 1.875×10^{-6} ，这与“的电影”出现的真实概率很接近——真实概率约为 1.6×10^{-5} 次方，是预测值的 8.5 倍。计算结果表明，“电影院”更可能是一个有意义的搭配，而“的电影”则更像是“的”和“电影”这两个成分偶然拼到一起的。

当然，作为一个无知识库的抽词程序，我们并不知道“电影院”是“电影”加“院”得来的，也并不知道“的电影”是“的”加上“电影”得来的。错误的切分方法会过高地估计该片段的凝合程度。如果我们把“电影院”看作是“电”加“影院”所得，由此得到的凝合程度会更高一些。因此，为了算出一个文本片段的凝合程度，我们需要枚举它的凝合方式——这个文本片段是由哪两部分组合而来的。令 p(x) 为文本片段 x 在整个语料中出现的概率，那么我们定义“电影院”的凝合程度就是 p(电影院) 与 p(电) · p(影院) 比值和 p(电影院) 与 p(电影) · p(院) 的比值中的较小值，“的电影”的凝合程度则是 p(的电影) 分别除以 p(的) · p(电影) 和 p(的电) · p(影) 所得的商的较小值。

可以想到，凝合程度最高的文本片段就是诸如“蝙蝠”、“蜘蛛”、“彷徨”、“忐忑”、“玫瑰”之类的词了，这些词里的每一个字几乎总是会和另一个字同时出现，从不在其他场合中使用。

光看文本片段内部的凝合程度还不够，我们还需要从整体来看它在外部的表现。考虑“被子”和“辈子”这两个片段。我们可以说“买被子”、“盖被子”、“进被子”、“好被子”、“这被子”等等，在“被子”前面加各种字；但“辈子”的用法却非常固定，除了“一辈子”、“这辈子”、“上辈子”、“下辈子”，基本上“辈子”前面不能加别的字了。“辈子”这个文本片段左边可以出现的字太有限，以至于直觉上我们可能会认为，“辈子”并不单独成词，真正成词的其实是“一辈子”、“这辈子”之类的整体。可见，文本片段的自由运用程度也是判断它是否成词的重要标准。如果一个文本片段能够算作一个词的话，它应该能够灵活地出现在各种不同的环境中，具有非常丰富的左邻字集合和右邻字集合。

“信息熵”是一个非常神奇的概念，它能够反映知道一个事件的结果后平均会给你带来多大的信息量。如果某个结果的发生概率为 p ，当你知道它确实发生了，你得到的信息量就被定义为

– log(p) 。 p 越小，你得到的信息量就越大。如果一颗骰子的六个面分别是 1 、 1 、 1 、 2 、 2 、 3 ，那么你知道了投掷的结果是 1 时可能并不会那么吃惊，它给你带来的信息量是 – log(1/2) ，约为 0.693 。知道投掷结果是 2 ，给你带来的信息量则是 – log(1/3) ≈ 1.0986 。知道投掷结果是 3 ，给你带来的信息量则有 – log(1/6) ≈ 1.79 。但是，你只有 1/2 的机会得到 0.693 的信息量，只有 1/3 的机会得到 1.0986 的信息量，只有 1/6 的机会得到 1.79 的信息量，因而平均情况下你会得到 0.693/2 + 1.0986/3 + 1.79/6 ≈ 1.0114 的信息量。这个 1.0114 就是那颗骰子的信息熵。现在，假如某颗骰子有 100 个面，其中 99 个面都是 1 ，只有一个面上写的 2 。知道骰子的抛掷结果是 2 会给你带来一个巨大无比的信息量，它等于 – log(1/100) ，约为 4.605 ； 但你只有百分之一的概率获取到这么大的信息量，其他情况下你只能得到 – log(99/100) ≈ 0.01005 的信息量。平均情况下，你只能获得 0.056 的信息量，这就是这颗骰子的信息熵。再考虑一个最极端的情况：如果一颗骰子的六个面都是 1 ，投掷它不会给你带来任何信息，它的信息熵为 – log(1) = 0 。什么时候信息熵会更大呢？换句话说，发生了怎样的事件之后，你最想问一下它的结果如何？直觉上看，当然就是那些结果最不确定的事件。没错，信息熵直观地反映了一个事件的结果有多么的随机。

我们用信息熵来衡量一个文本片段的左邻字集合和右邻字集合有多随机。考虑这么一句话“吃葡萄不吐葡萄皮不吃葡萄倒吐葡萄皮”，“葡萄”一词出现了四次，其中左邻字分别为 {吃, 吐, 吃, 吐} ， 右邻字分别为 {不, 皮, 倒, 皮} 。根据公式，“葡萄”一词的左邻字的信息熵为 – (1/2) · log(1/2) – (1/2) · log(1/2) ≈ 0.693 ， 它的右邻字的信息熵则为 – (1/2) · log(1/2) – (1/4) · log(1/4) – (1/4) · log(1/4) ≈ 1.04 。可见，在这个句子中，“葡萄”一词的右邻字更加丰富一些。

在人人网用户状态中，“被子”一词一共出现了 956 次，“辈子”一词一共出现了 2330 次，两者的右邻字集合的信息熵分别为 3.87404 和 4.11644 ， 数值上非常接近。但“被子”的左邻字用例非常丰富：用得最多的是“晒被子”，它一共出现了 162 次；其次是“的被子”，出现了 85 次；接下来分别是“条被子”、“在被子”、“床被子”，分别出现了 69 次、 64 次和 52 次；当然，还有“叠被子”、“盖被子”、“加被子”、“新被子”、“掀被子”、“收被子”、“薄被子”、“踢被子”、“抢被子”等 100 多种不同的用法构成的长尾……所有左邻字的信息熵为 3.67453 。但“辈子”的左邻字就很可怜了， 2330 个“辈子”中有 1276 个是“一辈子”，有 596 个“这辈子”，有 235 个“下辈子”，有 149 个“上辈子”，有 32 个“半辈子”，有 10 个“八辈子”，有 7 个“几辈子”，有 6 个“哪辈子”，以及“n 辈子”、“两辈子”等 13 种更罕见的用法。所有左邻字的信息熵仅为 1.25963 。因而，“辈子”能否成词，明显就有争议了。“下子”则是更典型的例子， 310 个“下子”的用例中有 294 个出自“一下子”， 5 个出自“两下子”， 5 个出自“这下子”，其余的都是只出现过一次的罕见用法。事实上，“下子”的左邻字信息熵仅为 0.294421 ， 我们不应该把它看作一个能灵活运用的词。当然，一些文本片段的左邻字没啥问题，右邻字用例却非常贫乏，例如“交响”、“后遗”、“鹅卵”等，把它们看作单独的词似乎也不太合适。我们不妨就把一个文本片段的自由运用程度定义为它的左邻字信息熵和右邻字信息熵中的较小值。

在实际运用中你会发现，文本片段的凝固程度和自由程度，两种判断标准缺一不可。只看凝固程度的话，程序会找出“巧克”、“俄罗”、“颜六色”、“柴可夫”等实际上是“半个词”的片段；只

看自由程度的话，程序则会把“吃了一顿”、“看了一遍”、“睡了一晚”、“去了一趟”中的“了一”提取出来，因为它的左右邻字都太丰富了。

我们把文本中出现过的所有长度不超过 d 的子串都当作潜在的词（即候选词，其中 d 为自己设定的候选词长度上限，我设定的值为 5 ），再为出现频数、凝固程度和自由程度各设定一个阈值，然后只需要提取出所有满足阈值要求的候选词即可。为了提高效率，我们可以把语料全文视作一整个字符串，并对该字符串的所有后缀按字典序排序。下表就是对“四是四十是十十四是十四四十是四十”的所有后缀进行排序后的结果。实际上我们只需要在内存中存储这些后缀的前 d + 1 个字，或者更好地，只储存它们在语料中的起始位置。

十
十十四是十四四十是四十
十是十十四是十四四十是四十
十是四十
十四是十四四十是四十
十四四十是四十
是十十四是十四四十是四十
是十四四十是四十
是四十
是四十是十十四是十四四十是四十
四十
四十是十十四是十四四十是四十
四十是四十
四是十四四十是四十
四是四十是十十四是十四四十是四十
四四十是四十

这样的话，相同的候选词便都集中在了一起，从头到尾扫描一遍便能算出各个候选词的频数和右邻字信息熵。将整个语料逆序后重新排列所有的后缀，再扫描一遍后便能统计出每个候选词的左邻字信息熵。另外，有了频数信息后，凝固程度也都很好计算了。这样，我们便得到了一个无需任何知识库的抽词算法，输入一段充分长的文本，这个算法能以大致 $O(n \cdot \log n)$ 的效率提取出可能的词来。

对不同的语料进行抽词，并且按这些词的频数从高到低排序。你会发现，不同文本的用词特征是非常明显的。下面是对《西游记》上册的抽词结果：

行者、师父、三藏、八戒、大圣、菩萨、悟空、怎么、和尚、唐僧、老孙、溃骸、什么、沙僧、太宗、徒弟、袈裟、妖精、玉帝、今日、兄弟、公主、玄奘、陛下、宝贝、性命、晓得、门外、妖魔、光蕊、观音、花果山、土地、木叉、东土、变化、变做、伯钦、判官、多少、真君、齐天大圣、蟠桃、丞相、魏征、扯住、溃骸澳、抬头、揭谛、言语、猪八戒、兵器、吩咐、安排、叩头、清风、哪吒、左右、美猴王、钉钯、孩儿、女婿、金箍棒、二郎、东西、许多、奈何、人参果、收拾、近前、太保、明月、南海、水帘洞、门首、弼马温、李天王……

《资本论》全文：

商品、形式、货币、我们、过程、自己、机器、社会、部分、表现、没有、流通、需要、增加、已经、交换、关系、先令、积累、必须、英国、条件、发展、麻布、儿童、进行、提高、消费、减少、任何、手段、职能、土地、特殊、实际、完全、平均、直接、随着、简单、规律、市场、增长、上衣、决定、什么、制度、最后、支付、许多、虽然、棉纱、形态、棉花、法律、绝对、提供、扩大、独立、世纪、性质、假定、每天、包含、物质、家庭、规模、考察、剥削、经济学、甚至、延长、财富、纺纱、购买、开始、代替、便士、怎样、降低、能够、原料、等价物……

《圣经》全文：

以色列、没有、自己、一切、面前、大卫、知道、什么、犹大、祭司、摩西、看见、百姓、吩咐、埃及、听见、弟兄、告诉、基督、已经、先知、扫罗、父亲、雅各、永远、攻击、智慧、荣耀、临到、洁净、离开、怎样、平安、律法、支派、许多、门徒、打发、好像、仇敌、原文作、名叫、巴比伦、今日、首领、旷野、所罗门、约瑟、两个、燔祭、法老、衣服、脱离、二十、公义、审判、十二、亚伯拉罕、石头、聚集、按着、祷告、罪孽、约书亚、事奉、指着、城邑、进入、彼此、建造、保罗、应当、摩押、圣灵、惧怕、应许、如今、帮助、牲畜……

《时间简史》全文：

黑洞、必须、非常、任何、膨胀、科学、预言、太阳、观察、定律、运动、事件、奇点、坍缩、问题、模型、方向、区域、知道、开始、辐射、部分、牛顿、产生、夸克、无限、轨道、解释、边界、甚至、自己、类似、描述、最终、旋转、爱因斯坦、绕着、什么、效应、表明、温度、研究、收缩、吸引、按照、完全、增加、开端、基

本、计算、结构、上帝、进行、已经、发展、几乎、仍然、足够、影响、初始、科学家、事件视界、第二、改变、历史、世界、包含、准确、证明、导致、需要、应该、至少、刚好、提供、通过、似乎、继续、实验、复杂、伽利略……

哦，对了，还有我最喜欢的，《人民日报》 2000 年 4 月新闻版的抽词结果：

发展、我们、经济、主席、江泽民、领导、建设、关系、教育、干部、企业、问题、主义、政治、群众、改革、政府、思想、加强、台湾、地区、北京、总统、世界、记者、代表、民族、组织、历史、访问、原则、努力、管理、今天、技术、市场、世纪、坚持、社会主义、财政、江泽民主席、增长、积极、精神、同志、双方、自己、友好、领导干部、进一步、基础、提高、必须、不断、制度、政策、解决、取得、表示、活动、支持、通过、研究、没有、学习、稳定、举行、欢迎、农村、生活、促进、科技、投资、科学、环境、领域、公司、情况、充分……

当然，我也没有忘记对人人网用户状态进行分析——人人网用户状态中最常出现的词是：

哈哈、什么、今天、怎么、现在、可以、知道、喜欢、终于、这样、觉得、因为、如果、感觉、开始、回家、考试、老师、幸福、朋友、时间、发现、东西、快乐、为什么、睡觉、生活、已经、希望、最后、各种、状态、世界、突然、手机、其实、那些、同学、孩子、尼玛、木有、然后、以后、学校、所以、青年、晚安、原来、电话、加油、果然、学习、中国、最近、应该、需要、居然、事情、永远、特别、北京、他妈、伤不起、必须、呵呵、月亮、毕业、问题、谢谢、英语、生日快乐、工作、虽然、讨厌、给力、容易、上课、作业、今晚、继续、努力、有木有、记得……

事实上，程序从人人网的状态数据中一共抽出了大约 1200 个词，里面大多数词也确实都是标准的现代汉语词汇。不过别忘了，我们的目标是新词抽取。将所有抽出来的词与已有词库作对比，于是得到了人人网特有的词汇（同样按频数从高到低排序）：

尼玛、伤不起、给力、有木有、挂科、坑爹、神马、淡定、老爸、卧槽、牛逼、肿么、苦逼、无语、微博、六级、高数、选课、悲催、基友、蛋疼、很久、人人网、情何以堪、童鞋、哇咔咔、脑残、吐槽、猥琐、奶茶、我勒个去、刷屏、妹纸、胃疼、飘过、考研、弱爆了、太准了、搞基、忽悠、羡慕嫉妒恨、手贱、柯南、狗血、秒杀、装逼、真特么、碎觉、奥特曼、内牛满面、斗地主、腾讯、灰常、偶遇、拉拉、

屌丝、九把刀、高富帅、阿内尔卡、魔兽世界、线代、三国杀、林俊杰、速速、臭美、花痴……

我还想到了更有意思的玩法。为什么不拿每一天状态里的词去和前一天的状态作对比，从而提取出这一天里特有的词呢？这样一来，我们就能从人人网的用户状态中提取出每日热点了！从手里的数据规模看，这是完全有可能的。我选了 12 个比较具有代表性的词，并列出了它们在 2011 年 12 月 13 日的用户状态中出现的频数（左列的数），以及 2011 年 12 月 14 日的用户状态中出现的频数（右列的数）：

下雪	33	92
那些年	139	146
李宇春	1	4
看见	145	695
魔兽	23	20
高数	82	83
生日快乐	235	210
今天	1416	1562
北半球	2	18
脖子	23	69
悲伤	61	33
电磁炉	0	3

大家可以从直觉上迅速判断出，哪些词可以算作是 12 月 14 日的热词。比方说，“下雪”一词在 12 月 13 日只出现了 33 次，在 12 月 14 日却出现了 92 次，后者是前者的 2.8 倍，这不大可能是巧合，初步判断一定是 12 月 14 日真的有什么地方下雪了。“那些年”在 12 月 14 日的频数确实比 12 月 13 日更多，但相差并不大，我们没有理由认为它是当日的一个热词。

一个问题摆在了我们面前：我们如何去量化一个词的“当日热度”？第一想法当然是简单地看一看每个词的当日频数和昨日频数之间的倍数关系，不过细想一下你就发现问题了：它不能解决样本过少带来的偶然性。 12 月 14 日“李宇春”一词的出现频数是 12 月 13 日的 4 倍，这超过了“下雪”一词的 2.8 倍，但我们却更愿意相信“李宇春”的现象只是一个偶然。更麻烦的则是“电磁炉”一行， 12 月 14 日的频数是 12 月 13 日的无穷多倍，但显然我们也不能因此就认为“电磁炉”是 12 月 14 日最热的词。

忽略所有样本过少的词？这似乎也不太好，样本少的词也有可能真的是热词。比如“北半球”一词，虽然它在两天里的频数都很少，但这个 9 倍的关系确实不容忽视。事实上，人眼很容易看出哪些词真的是 12 月 14 日的热词：除了“下雪”以外，“看见”、“北半球”和“脖子”也应该

是热词。你或许坚信后三个词异峰突起的背后一定有什么原因（并且迫切地想知道这个原因究竟是什么），但却会果断地把“李宇春”和“电磁炉”这两个“异常”归结为偶然原因。你的直觉是对的—— 2011 年 12 月 14 日发生了极其壮观的双子座流星雨，此乃北半球三大流星雨之一。白天网友们不断转发新闻，因而“北半球”一词热了起来；晚上网友们不断发消息说“看见了”、“又看见了”，“看见”一词的出现频数猛增；最后呢，仰望天空一晚上，脖子终于出毛病了，于是回家路上一个劲儿地发“脖子难受”。

让计算机也能聪明地排除偶然因素，这是我们在数据挖掘过程中经常遇到的问题。我们经常需要对样本过少的项目进行“平滑”操作，以避免分母过小带来的奇点。这里，我采用的是一个非常容易理解的方法：一个词的样本太少，就给这个词的热度打折扣。为了便于说明，我们选出四个词为例来分析。

下表截取了前四个词，右边四列分别表示各词在 12 月 13 日出现的频数，在 12 月 14 日出现的频数，在两天里一共出现的总频数，以及后一天的频数所占的比重。第三列数字是前两列数字之和，第四列数字则是第二列数字除以第三列数字的结果。最后一列应该是一个 0 到 1 之间的数，它表明对应的词有多大概率出现在了 12 月 14 日这一天。最后一列可以看作是各词的得分。可以看到，此时“下雪”的得分低于“李宇春”，这是我们不希望看到的结果。“李宇春”的样本太少，我们想以此为缘由把它的得分拖下去。

下雪	33	92	125	0.736
那些年	139	146	285	0.512
李宇春	1	4	5	0.8
看见	145	695	840	0.827
（平均）			313.75	0.719

怎么做呢？我们把每个词的得分都和全局平均分取一个加权平均！首先计算出这四个词的平均总频数，为 313.75 ；再计算出这四个词的平均得分，为 0.719 。接下来，我们假设已经有 313.75 个人预先给每个词都打了 0.719 分，换句话说每个词都已经收到了 313.75 次评分，并且所有这 313.75 个评分都是 0.719 分。“下雪”这个词则还有额外的 125 个人评分，其中每个人都给了 0.736 分。因此，“下雪”一词的最终得分就是：

下雪	$(0.736 \times 125 + 0.719 \times 313.75) / (125 + 313.75) \approx 0.724$

类似地，其他几个词的得分依次为：

那些年	$(0.512 \times 285 + 0.719 \times 313.75) / (285 + 313.75) \approx 0.62$

李宇春	(0.8 × 5 + 0.719 × 313.75) / (5 + 313.75) ≈ 0.7202
看见	(0.827 × 840 + 0.719 × 313.75) / (840 + 313.75) ≈ 0.798

容易看出，此时样本越大的词，就越有能力把最终得分拉向自己本来的得分，样本太小的词，最终得分将会与全局平均分非常接近。经过这么一番调整，“下雪”一词的得分便高于了“李宇春”。实际运用中， 313.75 这个数也可以由你自己来定，定得越高就表明你越在意样本过少带来的负面影响。这种与全局平均取加权平均的思想叫做 Bayesian average ，从上面的若干式子里很容易看出，它实际上是最常见的平滑处理方法之一——分子分母都加上一个常数——的一种特殊形式。

利用之前的抽词程序抽取出人人网每一天内用户状态所含的词，把它们的频数都与前一天的作对比，再利用刚才的方法加以平滑，便能得出每一天的热词了。我手上的数据是人人网 2011 年 12 月上半月的数据，因此我可以得出从 12 月 2 日到 12 月 15 日的热词（选取每日前 5 名，按得分从高到低）。

2011-12-02：第一场雪、北京、金隅、周末、新疆
2011-12-03：荷兰、葡萄牙、死亡之组、欧洲杯、德国
2011-12-04：那些年、宣传、期末、男朋友、升旗
2011-12-05：教室、老师、视帝、体育课、质量
2011-12-06：乔尔、星期二、摄影、经济、音乐
2011-12-07：陈超、星巴克、优秀、童鞋、投票
2011-12-08：曼联、曼城、欧联杯、皇马、冻死
2011-12-09：保罗、月全食、交易、火箭、黄蜂
2011-12-10：变身、罗伊、穿越、皇马、巴萨
2011-12-11：皇马、巴萨、卡卡、梅西、下半场
2011-12-12：淘宝、阿内尔卡、双十二、申花、老师
2011-12-13：南京、南京大屠杀、勿忘国耻、默哀、警报
2011-12-14：流星雨、许愿、愿望、情人节、几颗
2011-12-15：快船、保罗、巴萨、昨晚、龙门飞甲

看来， 12 月 14 日果然有流星雨发生。

注意，由于我们仅仅对比了相邻两天的状态，因而产生了个别实际上是由工作日/休息日的区别造成的“热词”，比如“教室”、“老师”、“星期二”等。把这样的词当作热词可能并不太妥。结合上周同日的数据，或者干脆直接与之前整个一周的数据来对比，或许可以部分地解决这一问题。

事实上，有了上述工具，我们可以任意比较两段不同文本中的用词特点。更有趣的是，人人网状态的大多数发布者都填写了性别和年龄的个人信息，我们为何不把状态重新分成男性和女性两组，或者 80 后和 90 后两组，挖掘出不同属性的人都爱说什么？要知道，在过去，这样的问题需要进行大规模语言统计调查才能回答！然而，在互联网海量用户生成内容的支持下，我们可以轻而易举地挖掘出答案来。

我真的做了这个工作（基于另一段日期内的数据）。男性爱说的词有：

> 兄弟、篮球、男篮、米兰、曼联、足球、蛋疼、皇马、比赛、国足、超级杯、球迷、中国、老婆、政府、航母、踢球、赛季、股市、砸蛋、牛逼、铁道部、媳妇、国际、美国、连败、魔兽、斯内德、红十字、经济、腐败、程序、郭美美、英雄、民主、鸟巢、米兰德比、官员、内涵、历史、训练、评级、金融、体育、记者、事故、程序员、媒体、投资、事件、社会、项目、伊布、主义、决赛、操蛋、纳尼、领导、喝酒、民族、新闻、言论、和谐、农民、体制、城管……

下面则是女性爱说的词：

> 一起玩、蛋糕、加好友、老公、呜呜、姐姐、嘻嘻、老虎、讨厌、妈妈、呜呜呜、啦啦啦、便宜、减肥、男朋友、老娘、逛街、无限、帅哥、礼物、互相、奶茶、委屈、各种、高跟鞋、指甲、城市猎人、闺蜜、巧克力、第二、爸爸、宠物、箱子、吼吼、大黄蜂、狮子、胃疼、玫瑰、包包、裙子、游戏、遇见、嘿嘿、灰常、眼睛、各位、妈咪、化妆、玫瑰花、蓝精灵、幸福、陪我玩、任务、怨念、舍不得、害怕、狗狗、眼泪、温暖、面膜、收藏、李民浩、神经、土豆、零食、痘痘、戒指、巨蟹、晒黑……

下面是 90 后用户爱用的词：

> 加好友、作业、各种、乖乖、蛋糕、来访、卧槽、通知书、麻将、聚会、补课、欢乐、刷屏、录取、无限、互相、速度、一起玩、啦啦啦、晚安、求陪同、基友、美女、矮油、巨蟹、五月天、第二、唱歌、老虎、扣扣、啧啧、帅哥、哈哈哈、尼玛、便宜、苦逼、斯内普、写作业、劳资、孩纸、哎哟、炎亚纶、箱子、无聊、求来访、查分、上课、果断、处女、首映、屏蔽、混蛋、暑假、吓死、新东方、组队、下学期、陪我玩、打雷、妹纸、水瓶、射手、搞基、吐槽、同学聚会、出去玩、呜呜、白羊、表白、做作业、签名、姐姐、停机、伏地魔、对象、哈哈、主页、情侣、无压力、共同、摩羯、碎觉、肿么办……

下面则是 80 后用户爱用的词：

> 加班、培训、周末、工作、公司、各位、值班、砸蛋、上班、任务、公务员、工资、领导、包包、办公室、校内、郭美美、时尚、企业、股市、新号码、英国、常联系、实验室、论文、忙碌、项目、部门、祈福、邀请、招聘、顺利、朋友、红十字、男朋友、媒体、产品、标准、号码、存钱、牛仔裤、曼联、政府、简单、立秋、事故、伯明翰、博士、辞职、健康、销售、深圳、奶茶、搬家、实验、投资、节日快乐、坚持、规则、考验、生活、体制、客户、发工资、忽悠、提供、教育、处理、惠存、沟通、团购、缺乏、腐败、启程、红十字会、结婚、管理、环境、暴跌、服务、变形金刚、祝福、银行……

不仅如此，不少状态还带有地理位置信息，因而我们可以站在空间的维度对信息进行观察。这个地方的人都爱说些什么？爱说这个词的人都分布在哪里？借助这些包含地理位置的签到信息，我们也能挖掘出很多有意思的结果来。例如，对北京用户的签到信息进行抽词，然后对于每一个抽出来的词，筛选出所有包含该词的签到信息并按地理坐标的位置聚类，这样我们便能找出那些地理分布最集中的词。结果非常有趣：“考试”一词集中分布在海淀众高校区，“天津”一词集中出现在北京南站，“逛街”一词则全都在西单附近扎堆。北京首都国际机场也是一个非常特别的地点，“北京”、“登机”、“终于”、“再见”等词在这里出现的密度极高。

从全国范围来看，不同区域的人也有明显的用词区别。我们可以将全国地图划分成网格，统计出所有签到信息在各个小格内出现的频数，作为标准分布；然后对于每一个抽出来的词，统计出包含该词的签到信息在各个小格内出现的频数，并与标准分布进行对比（可以采用余弦距离等公式），从而找出那些分布最反常的词。程序运行后发现，这样的词还真不少。一些明显具有南北差异的词，分布就会与整个背景相差甚远。例如，在节假日的时候，“滑雪”一词主要在北方出现，“登山”一词则主要在南方出现。地方特色也是造成词语分布差异的一大原因，例如“三里屯”一词几乎只在北京出现，“热干面”一词集中出现在武汉地区，“地铁”一词明显只有个别城市有所涉及。这种由当地人的用词特征反映出来的真实的地方特色，很可能是许多旅游爱好者梦寐以求的信息。另外，方言也会导致用词分布差异，例如“咋这么”主要分布在北方地区，“搞不懂”主要分布在南方城市，“伐”则非常集中地出现在上海地区。当数据规模足够大时，或许我们能通过计算的方法，自动对中国的方言区进行划分。

其实，不仅仅是发布时间、用户年龄、用户性别、地理位置这四个维度，我们还可以对浏览器、用户职业、用户活跃度、用户行为偏好等各种各样的维度进行分析，甚至可以综合考虑以上维度，在某个特定范围内挖掘热点事件，或者根据语言习惯去寻找出某个特定的人群。或许这听上去太过理想化，不过我坚信，有了合适的算法，这些想法终究会被一一实现。

2012 年 8 月 10 日 / 算法, 统计, 语言学

发表评论

评论

昵称*

邮箱*

网站

Save my name, email, and website in this browser for the next time I comment.

验证* × 9 = 45

提交