Project 1

李林翼 * 朱祺 †

April 14, 2017

Contents

*计 43, 2014011361, limyik.li96@gmail.com
†计 43, 2014011336, zhu-q14@mails.tsinghua.edu.cn

1 数据预处理和可视化

1.1 新闻数据读入与建立数据框对象

第一步是数据的读取。此处为了后面便于处理，将读取的数据持久化，保存为 result/data.csv 文件。可以分为以下三步：确定要读取的新闻的属性；读取新闻到 data.frame；保存为.csv 文件。

1.1.1 确定要读取的新闻的属性

根据 proj1 的要求和 new_york_times_annotated_corpus.pdf 文件，选定以下属性读取：

docid 新闻唯一标识符，也是文档的名字

title 新闻的标题

categories 新闻的类别，使用 online_sections 属性。例子："Business; Technology"。

locations 新闻中提到的地点，使用 Locations 和 Online_Locations 属性。例子："NEW YORK, NY"。

day_of_month,month,year 发行日期，使用 publication_* 属性。例子：26; 06; 1995。

publication_date 发行日期，使用 Publication Date 属性。例子：19950627T000000。

body 新闻正文。

1.1.2 读取新闻到 data.frame

这一部分主要的函数 readDoc() 在 readDoc.R 中。使用了 XML 和 stringr 两个库辅助处理。属性不存在时标记为 NA。

1.1.3 数据持久化

主要的函数 readAll() 和 extractAll() 在 readDoc.R 中。读取目录下所有新闻，将 data.frame 写入 data.csv

1.2 对新闻全文进行预处理

tm 库中有很方便的函数可以进行预处理，包括去除标点符号、停用词、数字、空白字符，将大写字母都转化为小写，以及词干化处理。所有的这些处理都可以使用 tm_map() 函数，通过 map 的方式将转化函数应用到每一个文档语料上。主要函数 getCorpus() 在 process.R 中，返回 Corpus。

1.3 将新闻表示成 BagOfWords 向量

利用上一步得到的 Corpus，借助 DocumentTermMatrix 函数，可以得到文档-词条矩阵，每一行即是 BagOfWords 向量。

1.4 筛选出出现次数大于 100 的词并画 wordcloud

DocumentTermMatrix 得到的文档-词条矩阵通过 findFreqTerms 函数找出出现次数大于 100 的词。单词-频数表保存在 result/wordFrequency.csv 中。利用 wordcloud 函数绘制云图。实现在 process.R 的 drawWordCloud() 中。结果见1。出现最多的词是 said，挺符合新闻报道的特点的。其他高频词如 state、compani、school、work、peopl、american 还是很合理的。但也有些没什么实际含义的词如 also、dont、next、get、just。

1.5 画出单词长度的分布直方图

与上类似，findFreqTerms(DocumentTermMatrix(corpus), 0) 得到所有 word，按单词长度统计。利用 ggplot2 中 qplot 画图。实现在 process.R 的 drawWordLength() 中。结果见2。可以看到单词的长度基本上在 10 以内，主要集中在 3-6 个字母之间。

1.6 画出新闻类别的分布直方图

将新闻按照类别进行统计。每个新闻可能有多个类别，要在这些类别下都计数。实现在 process.R 的 drawCategories() 中。利用 ggplot() 画图，结果见3。最多的是 business, new york, region，其他的如 art, sports, opinions 也不少，偏向生活的类别含有的新闻比较少，如 theater, travel, dining, automobile。

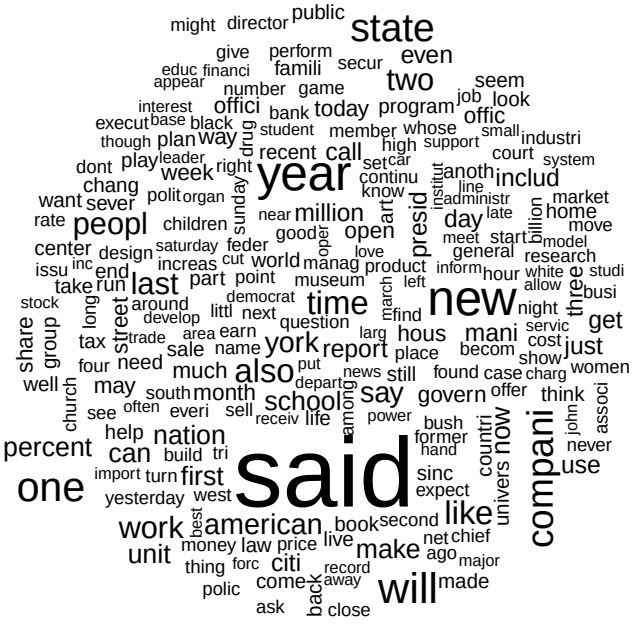

Figure 1: wordCloud

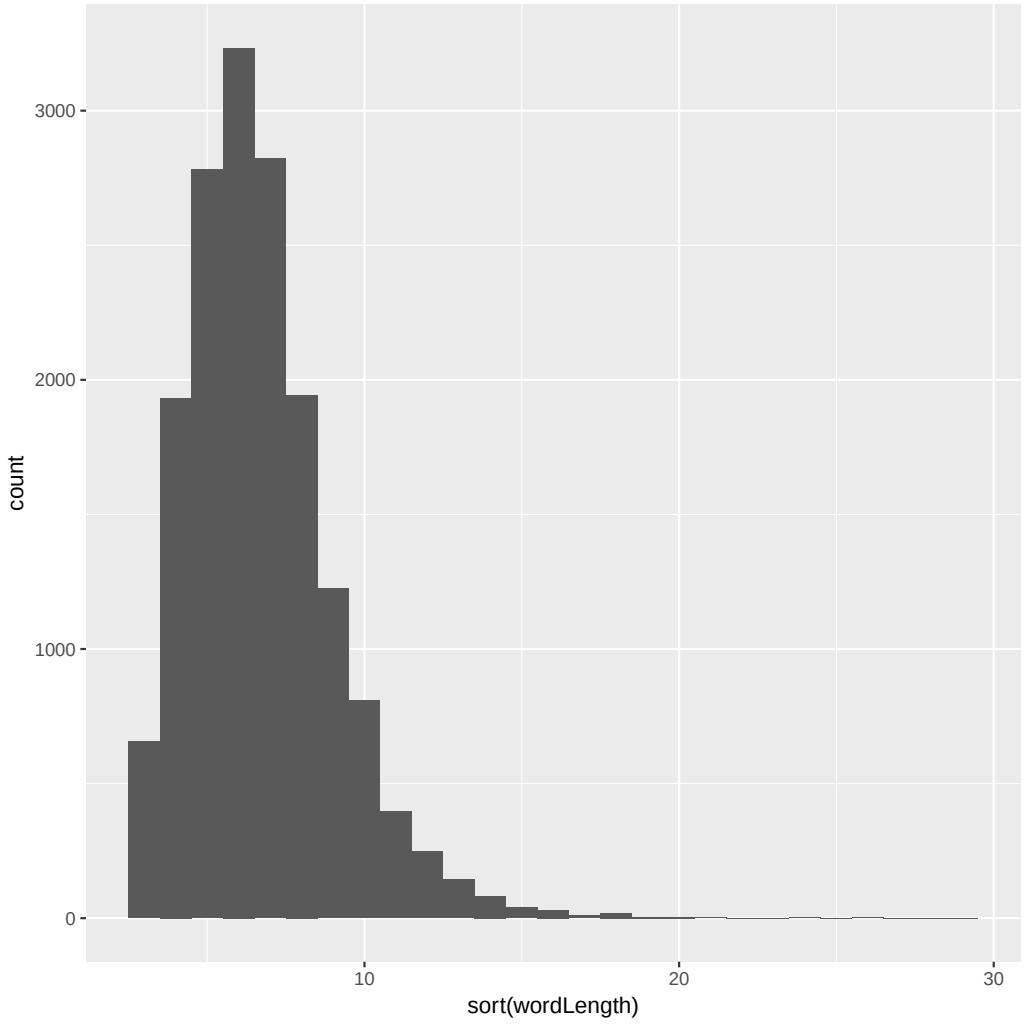

Figure 2: wordLength

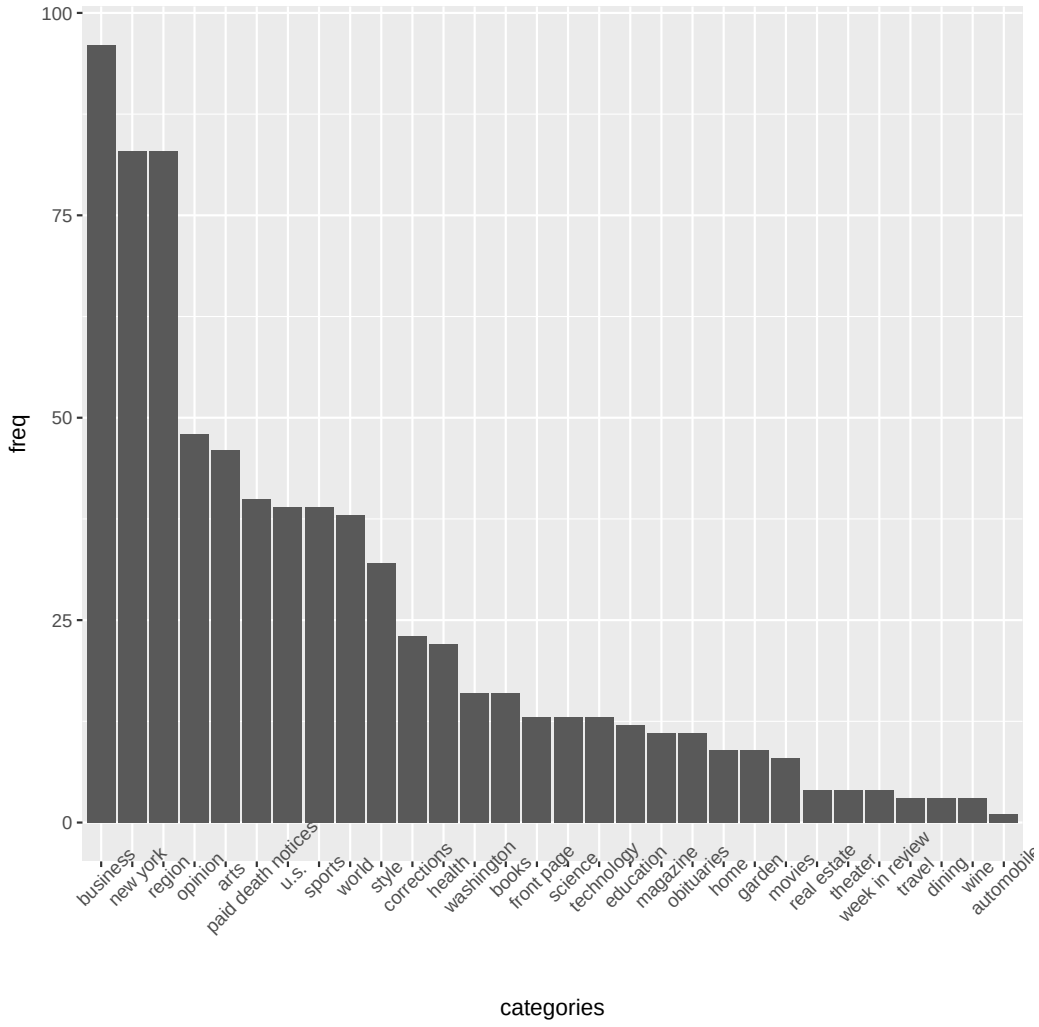

Figure 3: categories

1.7 画出每个月新闻数量的分布直方图

按月绘制新闻的出版数量，从所有新闻中找出最早的出版月份作为第 0 个月，其他出版月份以此为参照递推，次年 1 月为第 13 个月。实现在 process.R 的 drawTimeLine() 中。利用 ggplot() 画图，结果见 4。

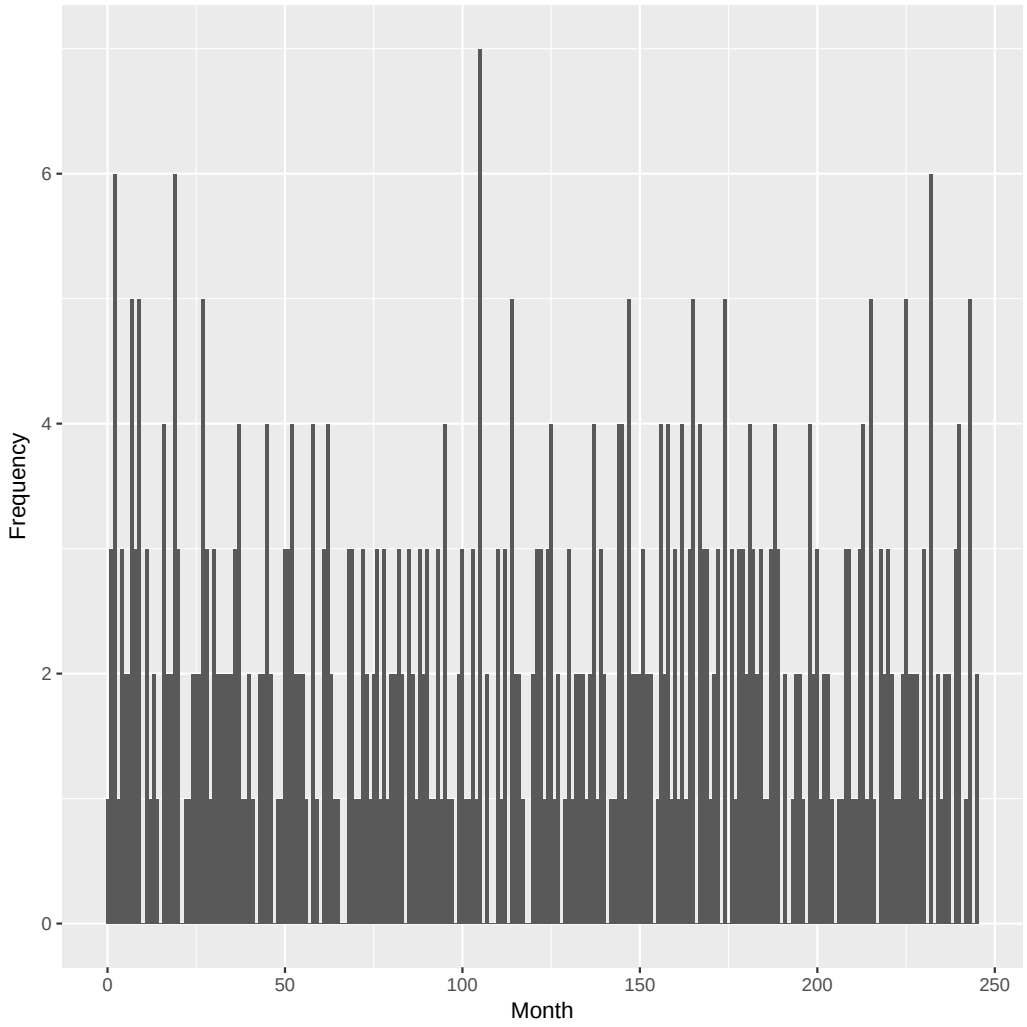

Figure 4: timeLine

1.8 画出每年新闻数量的分布直方图

由于每个月新闻数量的分布直方图太过密集，于是稍微修改 drawTimeLine() 函数得到 drawTimeLineYear() 函数，用于画出每年新闻数量的分布直方图。见5。可以看出都在 20 上下，波动不大。

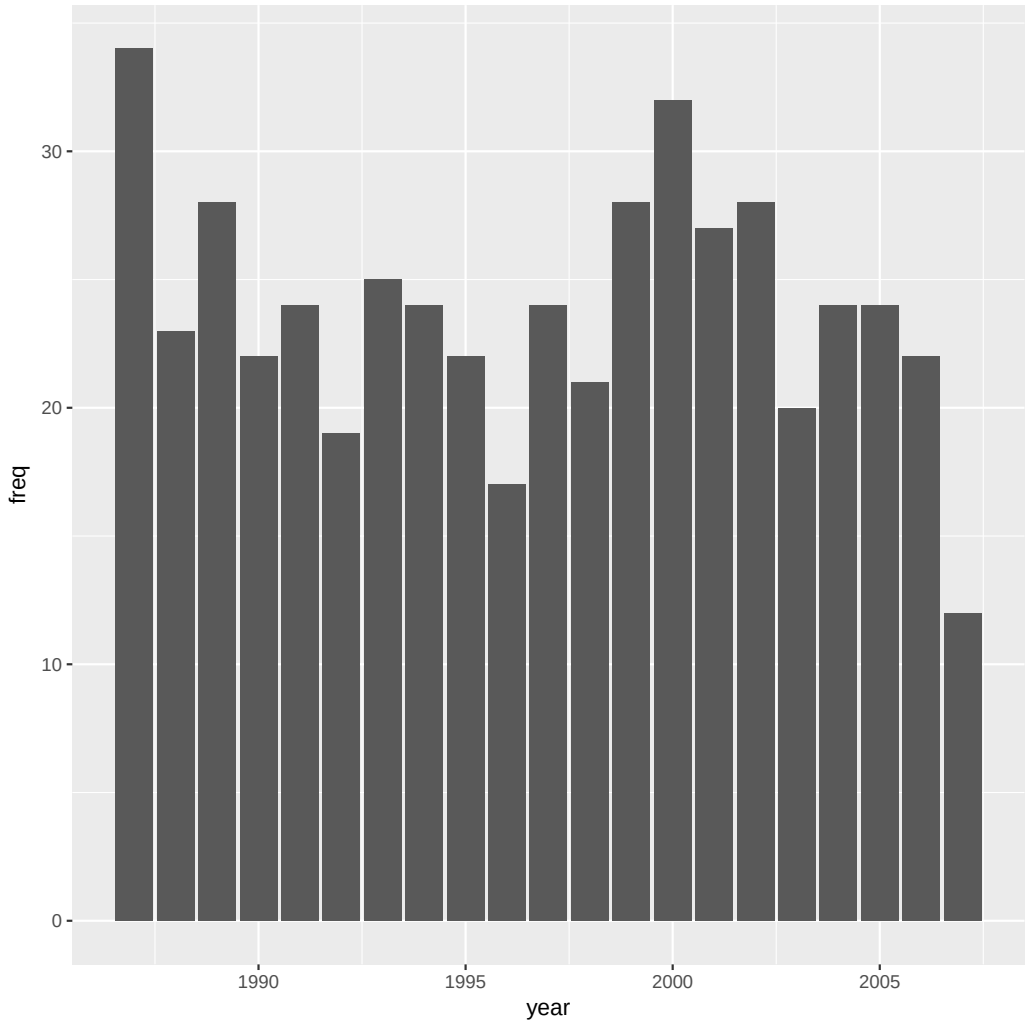

Figure 5: timeLineYear

2 新闻相似度计算

通过 tm 中函数 DocumentTermMatrix(corpus) 得到文档-词语矩阵，其中每一行就是新闻的 BoW 向量表示。相关函数 getWordMatrix()。

2.1 计算新闻之间的余弦相似度矩阵

用 BoW 向量计算新闻之间的余弦相似度，时间复杂度 $O(mn^2)$，m 为向量长度 (总词数)，n 为文档数。这一步非常耗时，主要原因在于 m 太大。因此在扩展分析部分尝试对此进行改进。主要函数 getSimilarityMat()，在 process.R 中，使用公式

$$cos\_sim(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}$$

计算余弦相似度矩阵的上三角部分，由对称性下三角可得，对角线置为 1。结果在 result/similarityMatrix.csv。

2.2 计算类别内新闻之间的平均相似度

有了相似度矩阵和类别标签，就能计算类别 i 和类别 j 新闻之间的平均相似度。形式化表示如下：

$$avg\_sim(i,j) = \begin{cases} mean_{ii \in i, jj \in j}(sim(ii, jj)) & j \neq i \\ mean_{ii \in i, jj \in j, ii \neq jj}(sim(ii, jj)) & j = i \end{cases}$$

不同类别间的相似度就是两个类别所有新闻 pair 的相似度取平均。类别内新闻相似度是除了相同新闻间的相似度取平均。getCrossDistance() 计算得到类别间平均相似度矩阵，其中对角元素就是类别内新闻之间的平均相似度。结果见 result/innerDistance.csv，使用 drawInnerDistance() 作图，见6。其中 automobile 为 1 是因为只有一篇相关新闻。

2.3 计算两个类别的新闻之间的平均相似度

类别间平均相似度矩阵见 result/relativityMatrix.csv。queryDistance() 可用于查询两个类别新闻之间的平均相似度。

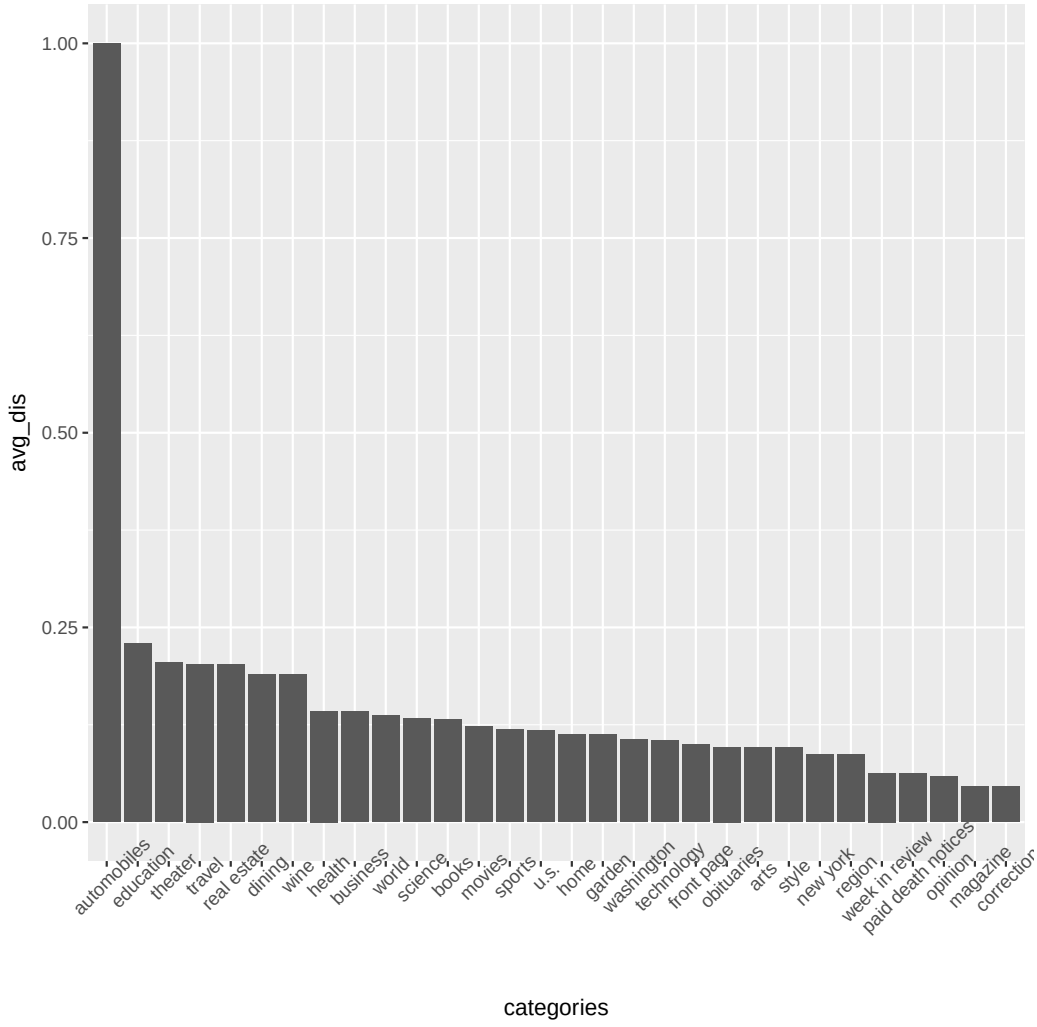

Figure 6: innerDistance

3 扩展分析

针对 BoW 向量太长的问题，想借助 SVD 进行降维处理。在 drawSVDrepr() 中，将文档-词语矩阵通过 SVD 分解，得到文档语义相关矩阵 datau 和词项语义相关矩阵 datav。取矩阵的前两维，即是对恢复原矩阵帮助最大，保留信息最多的两维进行作图分析，降维后文档的分布如7，词语的分布如8。很难从中获得一些可解释的信息，猜测是降维降得太厉害了，二维的投影很难表现原来成百上千维的内容。

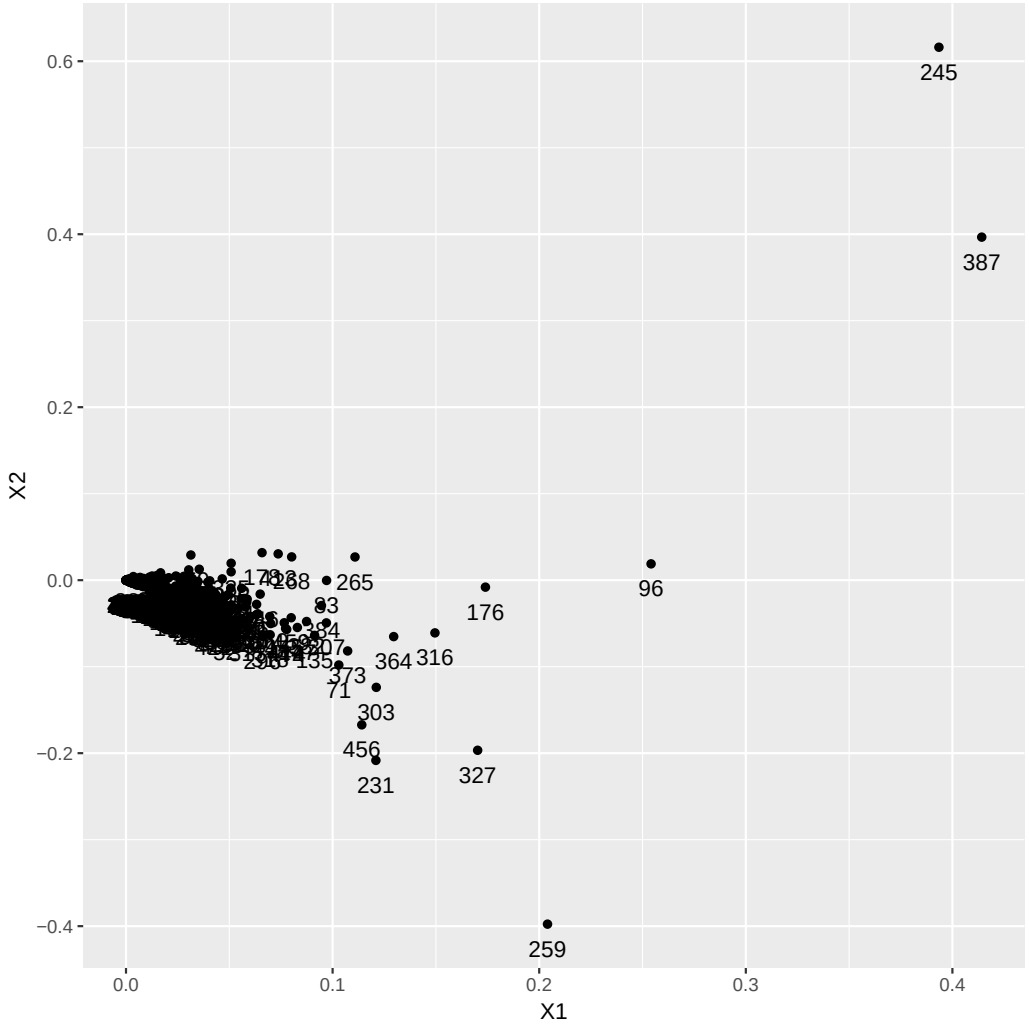

Figure 7: doc repr

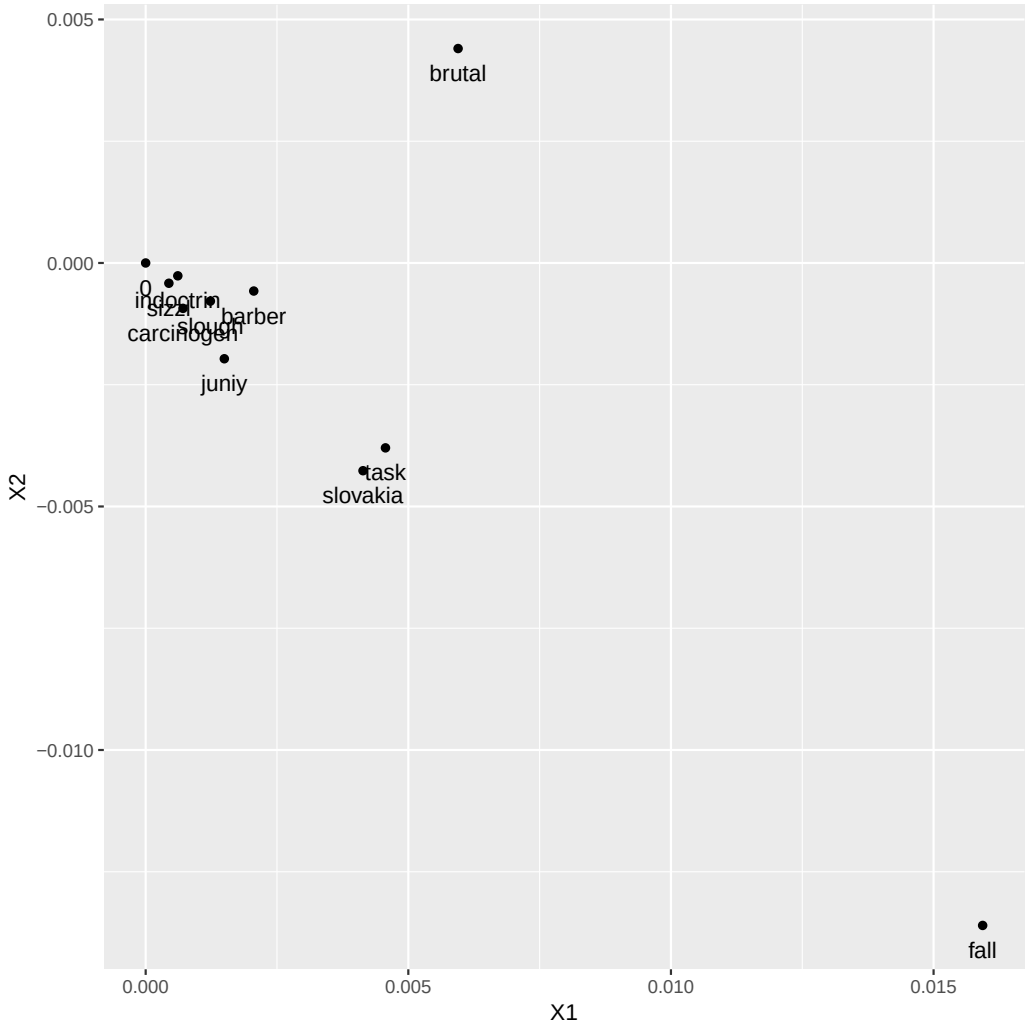

Figure 8: word repr