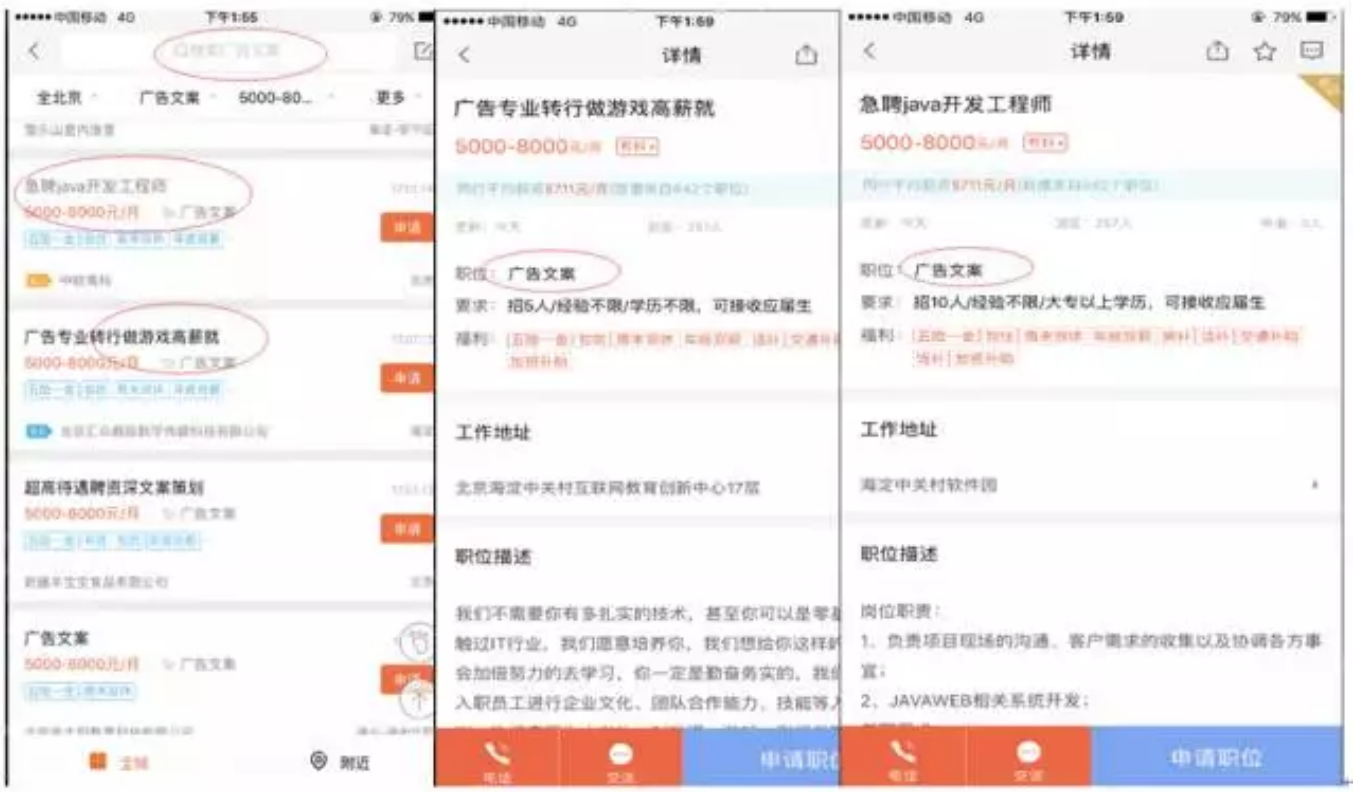
基于Word2Vector做文本分类

原创 锦煌@招聘技术 58招聘技术团队 2017-06-27

一、引子

笔者所在的数据策略团队近期在做推荐策略的时候发现了一些bad cases，即商户在发布职位的时候，有意或无意地错误发布本不属于某类目的职位。如下图所示，在广告文案类目中，我们可以发现里面包含游戏开发和java开发等类目。



首先，用户使用关键词进行搜索或者点击某类目进行搜索，说明有强烈的找该方面工作的意愿，一旦返回的结果中包含上述毫不相干的职位，将会极大地降低用户体验甚至使用户对产品可靠性产生质疑。

其次，上文提到了这可能是商户有意为之的结果。因为我们观察到存在某些灰产用户，其在几乎每个招聘二级类目下都发布了大量与当前类目毫不相干的计算机培训的职位。

综上，我们需要基于商户发布的职位标题/职位描述去进行文本分类，将商户发布的职位类目和我们分析的结果进行对比，对于相关性很低的（比如商户类目为程序员，我们分类结果是服务员），我们在展示页面给予降权，使相应帖子不优先展示或者不展示。

本文将提出一种基于Word2Vector模型的文本分类方法解决上述问题。行文逻辑如下：第二节介绍传统的做文本分类的方法；第三节介绍Word2Vector模型原理；第四节介绍我们如何使用该模型做职位的分类；第五节介绍踩过的一些坑。

本文旨在用较为通俗形象的文字解释算法，本人才疏学浅，如有不当之处，欢迎高手不吝赐教。

二、文本分类简介

2.1概述

文本分类，顾名思义，即给定类目体系下，根据文本的内容确定类别的过程。在我们的问题中，我们将使用58招聘的类目体系，将商户发布的职位标题及职位描述作为文本内容。

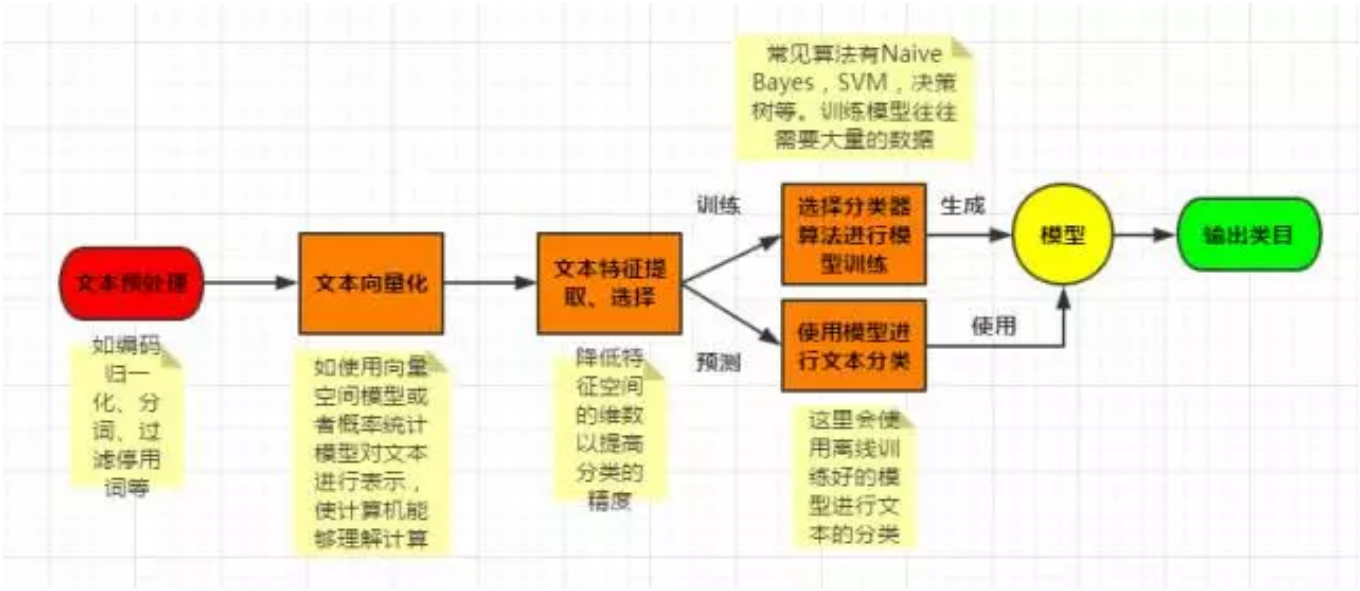
2.2历史及现状

1960-1970：人工+规则（关键词或者正则表达式）的方式，制定规则的人需要对某类目领域有足够的认知和了解。举个栗子：类目词是厨师，涉及到的规则可能是关于美食的、烹饪技巧的（刀工/炒锅/烘焙）或者厨师工种的（水台/砧板/打荷），需要涉及到方方面面，繁琐复杂。

1970-1990：信息检索概率模型（如向量空间模型Vector Space Model），以及相关评价指标如准确率、召回率的引入。

1990-2000：研究主要集中在了文本特征（即词语）的提取、选择以及分类器模型的设计方面。

目前业内做文本分类的主流方法基本上是用有监督的方式（即训练数据既含有文本数据，又带有相应的类目标签以帮助模型去学习该类目的表达），其过程一般如下图所示：



ps. 向量空间模型VSM：将文本抽象成一个向量，假设我们一共有N个词，那么一篇文本可以表示为(w1, w2, w3, … , wn)，其中wi表示第i个词在这篇文本中的权重，其计算方法一般用词频-逆文章频率即tf-idf来计算。

三、Word2Vector简介

3.1词向量模型

之所以要引入词向量模型，是因为上述传统的VSM存在如下缺点：

①一般情况下，词语的总数N会很大（注意N不仅仅是单篇文章的词语数，而是整个语料库的词语个数），而导致单篇文章的特征向量会很稀疏（即向量中大部分词语的权重都为0）。

②不能很好地刻画词与词之间的相似性。相似性是词语之间一个很重要的性质，举个极端的栗子：假设我们一共有3个词语：“招聘”、“美容师”、“化妆师”；有2篇文章：“招聘美容师”，“招聘化妆师”。两者语义很接近，然而其表示向量分别为[1,1,0]和[1,0,1]，我们可以测算一下余弦相似度，为1/2，即60°的夹角，说明两者“语义距离”很遥远。

目前业界一般用Distributed Representation方式的词向量模型，最早由Hinton在1984年提出。其将每一个词语映射成一个固定长度的“短”向量（比如50维，100维，相对于语料库动辄几十万的词语数量而言很短了）。类似于VSM，也是将每个词都当做词向量空间中的一点，在此空间中引入的距离概念，就是词语之间语义层面的相似性了。

3.2Word2Vector模型

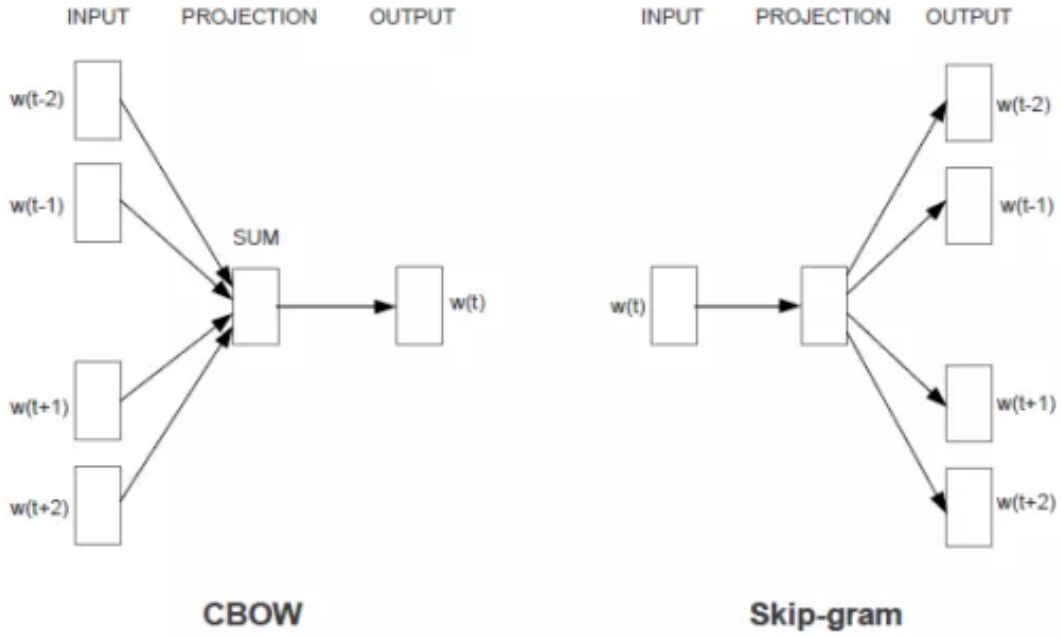
2013年，Google开源了一个用于生成词向量的工具，因其简单实用高效而引起广泛关注。其中具体的原理因篇幅原因，本文只稍作介绍，若有兴趣的读者，可阅读作者的原论文[8]（不涉及太多算法细节）、Word2Vec相关数学原理[9]进行了解。

Word2Vector本质上有两个学习任务，还有两套模型。

①两个学习任务分别是：

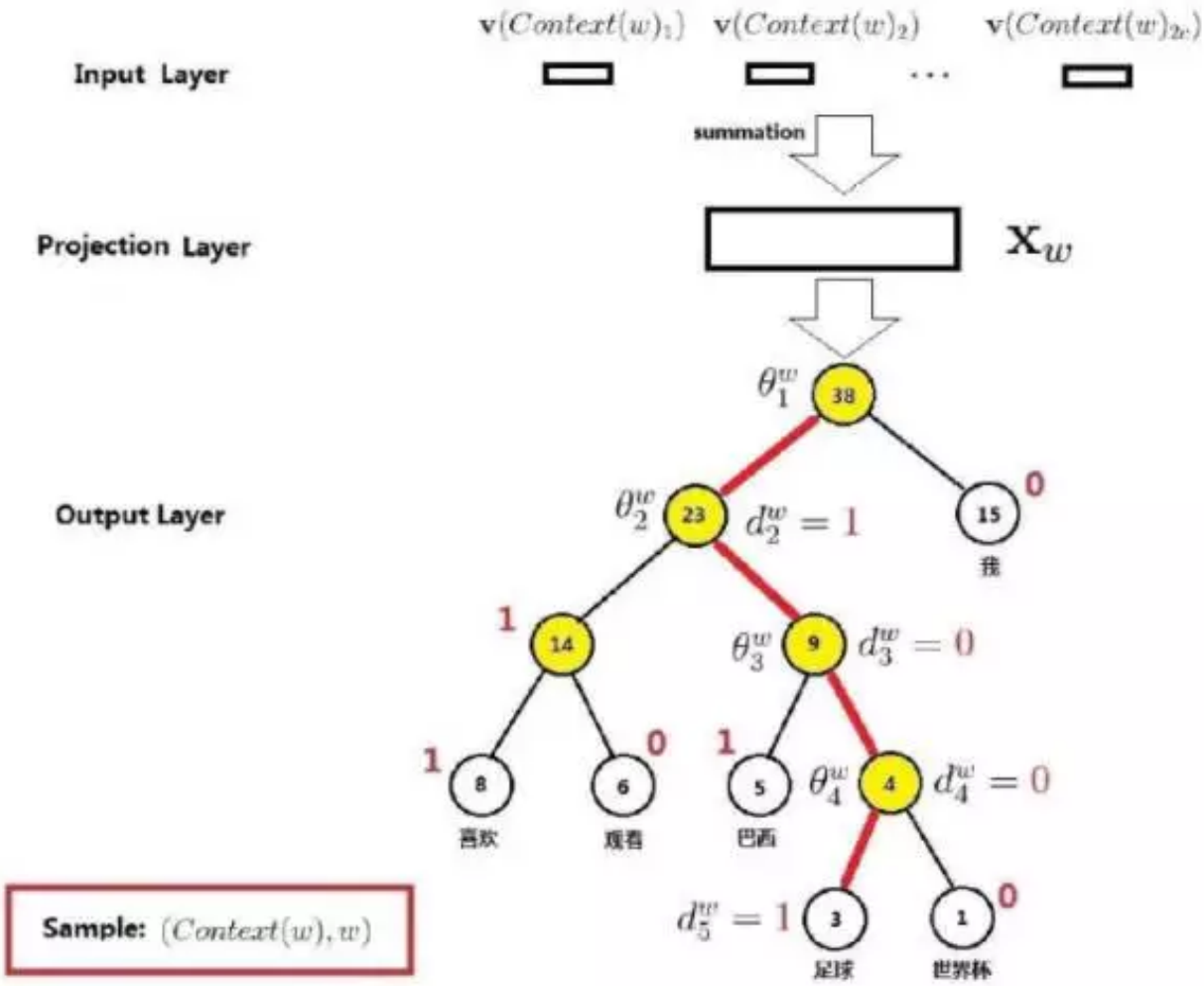
CBOW：对于每个词，用其周围的词，来预测该词生成的概率。

Skip-gram：对于每个词，用其自身去预测周围其他词生成的概率。

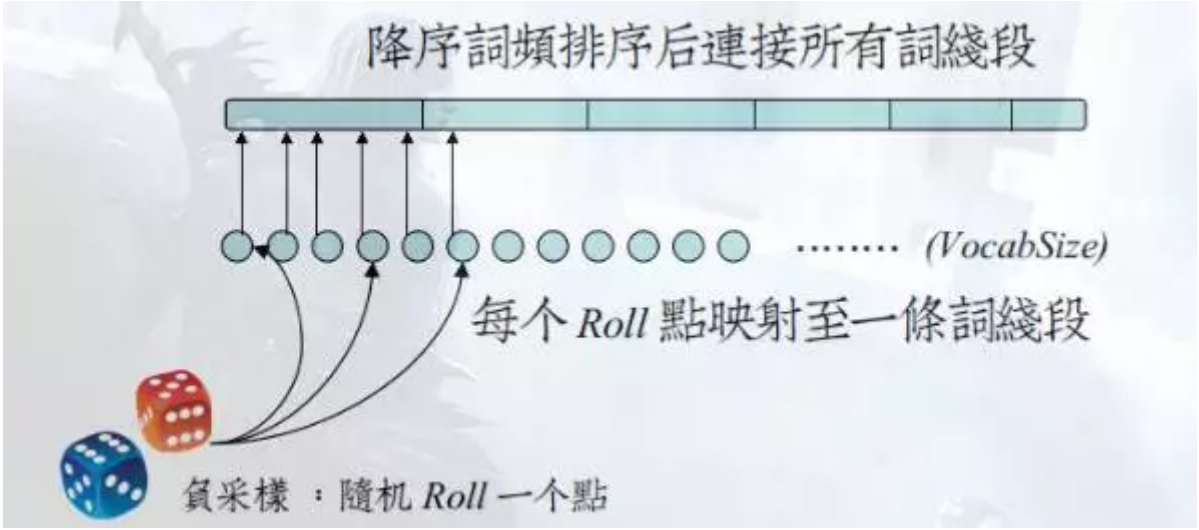


②两个模型分别是：

HierarchicalSoftmax模型：算法输入为初始随机的词向量，输出为一个包含语料里所有词语的Huffman树（最优二叉树），对于其中每一个叶节点（代表一个词），其从根节点到叶节点的路径即为此词语的Huffman编码。从名字上看，Softmax是用于解决多分类问题的（可以形象的理解为从所有词语中选择一个条件概率最大的），而此模型将一个多分类问题转化多个带有层级关系的二分类问题（这样能够将时间复杂度大大降低），即在每一个非叶子节点，都做一次二分类（输出为0或者1）。下图是该模型一个形象的展示（来自于[9]）：



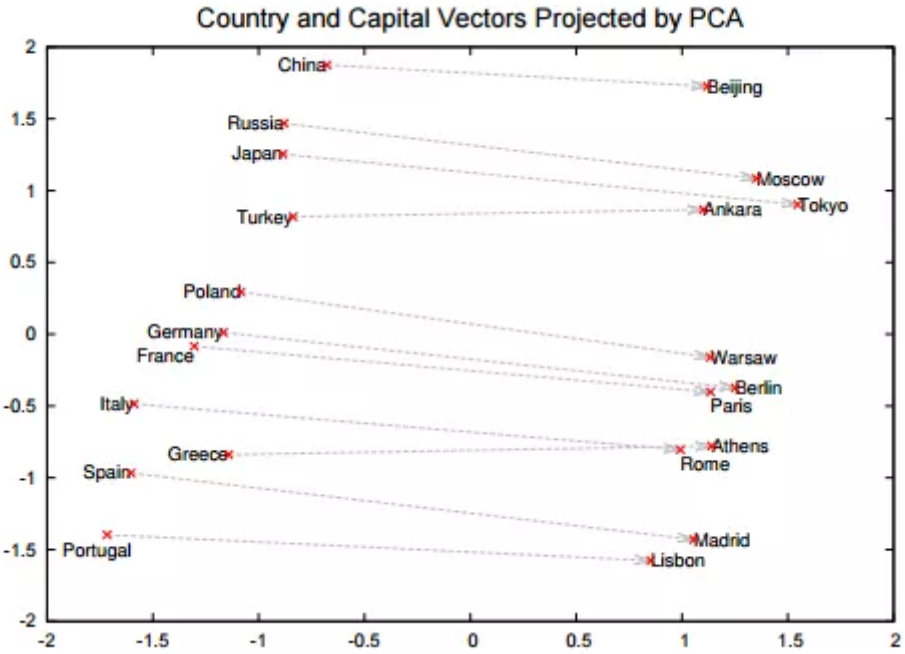
NegativeSampling模型：提出这个模型本质上是为了提高性能，不需要再生成复杂的Huffman树。负采样，顾名思义，关注点主要在负样本上，因为正样本是确定且数量很少的（比如CBOW里，给定context(w)，w就是唯一正样本）。我们需要的是带权采样，即对于高频词汇，我们希望选中的概率大；低频则较小。下图展示了Word2Vec中负采样的策略（来自于[10]）



3.3模型特性

前面也提到过，Word2Vec是用于解决词与词之间的相似性问题，更准确地说是在给定语境下，同时出现的概率（即词的共现性co-occurence）。举个栗子，我们选取最近一个月的新浪微博数据当做语料库去训练模型。那么当我输入“迪丽热巴”这个词的时候，模型返回最相似的词语可能是：“鹿晗”、“跑男”、“吃货”、“小姐姐”这些词（前提是我们分词的时候，这些词语能够被正常切分）。

除了上述相似性（共现性），Word2Vec作者还在[8]中提到了模型能自动将词语实体的概念组织起来并学习它们之间的隐含关系。下图是一个部分词向量经过PCA降维投影之后的展示，我们可以看到，首都词和国家词之间的语义距离几乎一样，而模型训练的过程中，我们并不需要提供任何此类信息，这完全是模型自己学习到的。



四、Word2Vec应用于文本分类

4.1可行性分析

第二节我们讨论了文本分类的一般步骤，基本上都是在大量有数据集上做的有监督学习，需要大量人工标注数据（58自己有商户和用户数据，其在录入过程中都会标上类目，且类目不准的占比很小，所以这方面问题不太大，而对于有些任务来说，标注数据获取成本极高）。而第三节我们看出，Word2Vec模型不需要标注数据，其能自发的学习出词与词之间的相似性和某些概念之间的内在联系。

那如何将其用做分类呢？

词向量，就是多维空间的一个向量，而两个词语的相似程度，可以由他们的向量的余弦距离（夹角）来描述。夹角越小，表明相似性越高，即越相似的词语指向的方向越一样。

类目体系由类目词语组成，假定我们有一个好的类目体系，类目词之间具有完备性和互斥性，那么不同类目词向量所指向的方向应该是不一样的。我们要做文本的分类，只要将文本映射到同一个词向量空间中，将文本也表示成某一个向量，然后看看这个向量和哪个类目词向量夹角最小，不就是这个文本属于的类目么？

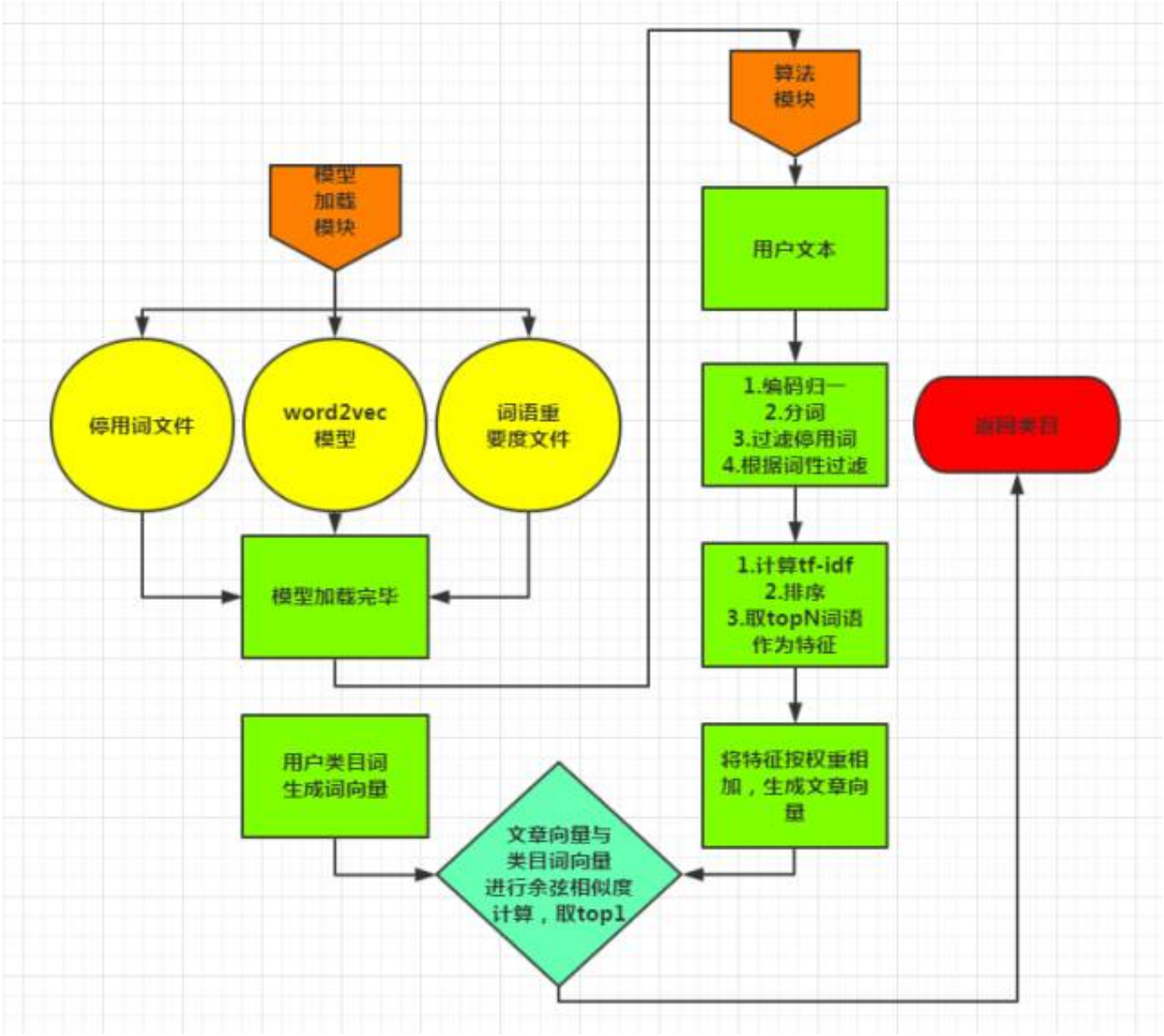
如何映射文本到词向量空间？

文本是由词语组成的，但是并不是每个词语对于文本的表述能力都是一样的。举个栗子，某职位信息是：“本工厂急招车床技工，要求有实际操作经验、年龄不限，学历不限”。最重要的几个词语（特征）应该是“工厂”、“车床”、“技工”这几个或代表了场所，或代表了技能，或代表了工种（实际上有个类目词就是技工）的名词性质的词语，其他词语都不能很好的帮助我们确定类目。

所以在由词语向量生成文章向量之前，我们还需要做一步特征选择，这和我们在第二节中提到的特征选择过程是类似的，仅仅是使用的方法有差异。

4.2算法详述

将4.1节的思路连贯的串起来，我们可以得到如下算法流程：



有如下几点说明：

①上图只是模型预测部分的文本分类流程，并不包含模型训练的部分。

②停用词指的是诸如“我”、“你”、“是”等出现在大量文章中，但是对文章的语义表达无任何作用的词语。词语重要度文件是离线计算获得的词语IDF（参见[6]）文件，包含语料库中任意词语以及其权重。

③模型加载完毕后，首先对用户文本进行预处理，包含了4个小步骤；其次进行特征选择，本算法中使用了tf-idf的方法对特征进行排序；然后计算文章向量的时候，需要根据特征的权重转化词语特征为文章特征；最后与类目词进行余弦相似度的计算，取相似度最高的类目词作为最终类目。

五、踩过的坑

1、笔者之前应用此方法进行过新闻文本的分类，取得过较好的效果。其类目体系中一级类目约20个（政治、体育、经济、娱乐等），每个一级类目下二级类目约20-30个（如体育下有篮球、足球、网球、乒乓球等），二级类目总数约为500多个。当时采取的方法是，先分一级类目，然后在当前一级类目中继续划分二级类目，这样能避免一次性划分500多个类目，极大程度上提高分类的精度。

然而这个方法并不能应用于58招聘的类目体系中。虽然其类目也是分层级的，且每一层级的个数并不多（30-40）。有以下几个原因：

①商户在写职位标题和职位描述的时候，倾向于直接用三级级类目相关描述词而不用二级类目的相关描述词（注：58招聘属于一级类目，所以这里从二级类目开始算起）（比如用“水泥工”、“砖瓦工”这样的三级类目细分词，而很少用“普工”、“技工”这样的二级类目抽象词），导致部分二级类目词语出现次数极低，这样算出的二级类目词和三级类目描述词的共现性肯定是极低的，从而导致分二级类目的时候准确率就不高，更别提分三级类目了。

②很多三级类目描述词词和好几个二级类目均有共现情况，如“宠物美容”和“医疗/医院/护理”以及“美容/美发”。这样算出共现性肯定会有偏差。

③类目的互斥性不够，如二级类目“汽车制造/服务”和“司机/交通服务”很相似。

④类目本身应该按照功能性区分，但是有的却按照级别去区分，比如二级类目存在“高级管理”和“生产管理/研发”这样的类目，那么销售总监究竟是应该分到“高级管理”还是应该分到“销售”？

⑤部分二级类目词太过于抽象：“美术/设计/创意”中的创意，“政府/非营利机构”中的非营利机构等。

笔者尝试对类目词进行过人工修改，可以提高部分效果，但是很费时费力。目前暂时采用的方法是直接进行三级类目的划分（1000多个），取top20，然后跟商户打上的三级类目进行对比，如果其类目不在这top20中，则对帖子采取降权处理。

2、对于文章特征的选择。这方面最大的感触就是，职位描述文本的质量实在是堪忧，表现为：

①大量重复性的千篇一律的描述，如对于薪酬福利的描述（日薪200，随走随结，上二休一等等），对于学历年龄等的要求（高中学历，30岁以下，服从领导安排等）。

②有价值的信息往往只在标题中或者在正文的某一个小角落。出于吸引点击的考虑，很多商户都是“标题党”，巴不得把整个职位所有的特点都在标题中描述完全。举个栗子：标题为“急招电工+五险+双休包吃住”，而正文只字不提或只提一句跟电工相关的东西。这种基本上根据标题就能选取特征“电工”直接进行分类了。

从上面能看出，只根据词性进行过滤，特征选择只根据tf-idf是不够的，必须要有一些小tricks。笔者用到或将要用到如下几点：

①目前选取特征主要是选取名词，但是更需要细分为“职位词”（股票交易员、货车司机等）、“场所词”（工厂、酒吧等）、“技能词”（java、c1等）并给予不同的权重予以区分。

②对于职位描述，尝试用正则表达式提取出“岗位职责”的描述部分，而忽略其余关于“岗位要求”、“薪酬待遇”等部分。

③标题相对于正文给予更大的权重。

3、对于正文长度过短的帖子，可以尝试直接对于标题去进行类目词的匹配。这样做的原因很简单：正文长度过短，特征数量不足以很好的表达文本。而标题中如果带有一模一样的类目词，那么直接做匹配，是更加精准的。

二、参考文献

[1]http://blog.csdn.net/chl033/article/details/4733647文本分类概述
[2]Lewis D D. Representation and learning ininformation retrieval[D]. University of Massachusetts, 1992.
[3]Yang Y, Pedersen J O. A comparative study onfeature selection in text categorization[C]//Icml. 1997, 97: 412-420.
[4] Freund Y, Schapire R E. Adesicion-theoretic generalization of on-line learning and an application toboosting[C]//European conference on computational learning theory. SpringerBerlin Heidelberg, 1995: 23-37.
[5] Jiang J, Wu C, Liang Y. Multi-categoryclassification by least squares support vector regression[J]. Advances inNeural Networks–ISNN 2005, 2005: 787-832.
[6] http://baike.baidu.com/link?url=v1dDZvWp5XCMi_iyGq734_vod7rQ4kHkKcGwlxGrXgyXi848g-fmUkNNmmQ0K_XvNmZcrsZVYrXvrPuiXNYdO_百度百科tf-idf
[7] Hinton G E. Distributed representations[J].1984.
[8] Mikolov T, Sutskever I, Chen K, et al.Distributed representations of words and phrases and theircompositionality[C]//Advances in neural information processing systems. 2013:3111-3119.
[9]http://blog.csdn.net/itplus/article/details/37969519 Word2Vec数学原理
[10]http://www.cnblogs.com/neopenx/p/4571996.html Word2Vec源码解析