LDA 主题模型及其应用研究 SRT 项目报告

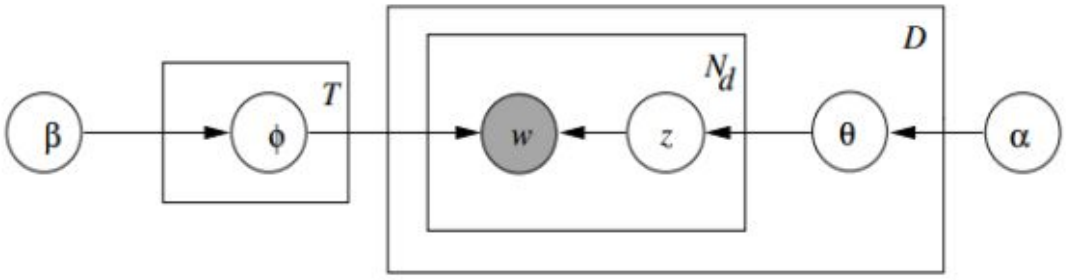
0.综述

本 SRT 的实验是做一个微博（主要是中文短文本）的挖掘系统，包括生成某一时刻某一地区发出的微博的主题摘要和提取关键词或短语。使用的语料集包括约 2 万条关于雅安地震和 44.7 万条关于钓鱼岛的微博。它们都是在特定的时间随机从网上抓取的含有对应关键字（“雅安地震”和“钓鱼岛”）的微博。

我先后学习了 LDA 及其变种、自动文摘生成算法和 Word Embedding 的相关知识。通过改进相关摘要生成算法以及融入主题模型（Topic Model）提出自己的方法。

1.LDA 主题模型简介

LDA(Latent Dirichlet Allocation)是一种文档主体生成模型,它包括 3 层结构——单词、主题、文档。其图模型如下，我们认为主题（记为 z）在文档（记为 d）上以及单词（记为 w）在主题上均服从多项分布；这两个多项分布的参数为 theta 和 phi，而这两个参数又分别服从超参数为 alpha 和 beta 的狄利克雷（Dirichlet）分布（因为 Dirichlet 分布是多项分布的共轭分布，计算迭代时候非常方便），超参数 alpha、beta 由人指定。



从上图可以看出，除了指定的 alpha 和 beta 外，只有 w 是可见的（标记为深色），z、theta、phi 是需要学习的。按照 LDA 的图模型，我们可以计算给定上述参数的情况下当前可见文档的后验概率如下，我们需要优化 z、theta 和 phi，使得后验概率（或其对数）尽可能高。通常情况下我们可以用吉布斯采样（Gibbs Sampling）来实现。

$$P(\theta,\phi,\vec{z},\vec{w}|\alpha,\beta)=P(\theta|\alpha)P(\phi|\beta)\prod_{n=1}^{N}P(z_n|\theta)P(w_n|z_n,\phi)$$

原始的 LDA 是一种基于共现的非监督学习方法，在自然语言中有比较多的应用。LDA 的一个良好特性是能够添加先验形成很多变种，例如优先未出现主题的非对称 LDA（ASLDA）和基于一阶逻辑表示先验的 LogicLDA。

在本次实验中我们只要运用 LDA 进行聚类操作，将为数众多的 Word 分为数量有限的 Topic，为后续的操作作准备。

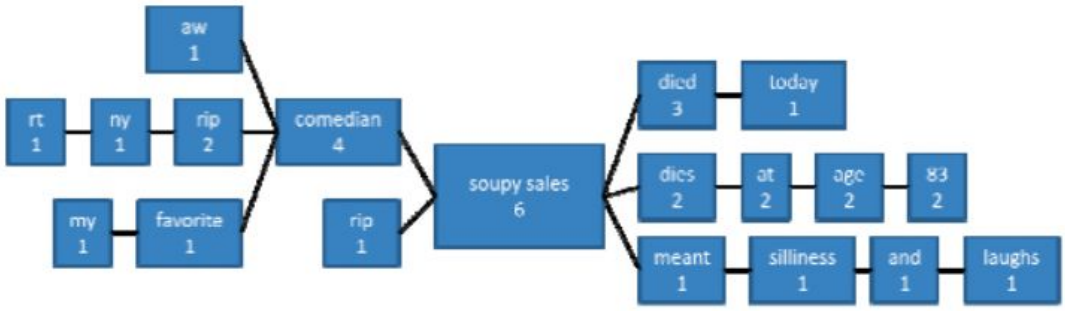
2.短文本摘要生成算法

本实验中我参考的摘要生成算法是基于语义树的，由于同一个语料集都含有同一个关键词，我们可以以此关键词为中心，向前向后生成两个语义树。这两个语义树都是以中心词为根，根节点对应整个语料集中的句子。每个节点的扩展方法如下：在该节点对应的句子中，设该节点对应的词前（后）一个词的集合元素数为 K，则该节点新建 K 个子节点，每一个子节点对应当前词前（后）一个词为 K[i]的句子。递归地进行上述这个过程，直到每个叶子节点对应的句子数均小于某一个先前指定好的阈值。在两个语义树建立完成后，我们会对每一个节点给一个权重，计算公式如下，其中 Count(Node)为该节点对应的句子的数量，Distance(Node)为该节点距离根节点的距离（根节点记为 0）。

$$Weight(Node) = Count(Node) - Distance(Node) * log_b\, Count(Node)$$

下面是一个例子，下面的六句话对应下侧蓝色的语义生成树。

Aw, Comedian Soupy Sales died.
RIP Comedian Soupy Sales dies at age 83.
My favorite comedian Soupy Sales died.
RT @NY: RIP Comedian Soupy Sales dies at age 83.
RIP: Soupy Sales Died Today.
Soupy Sales meant silliness and laughs.



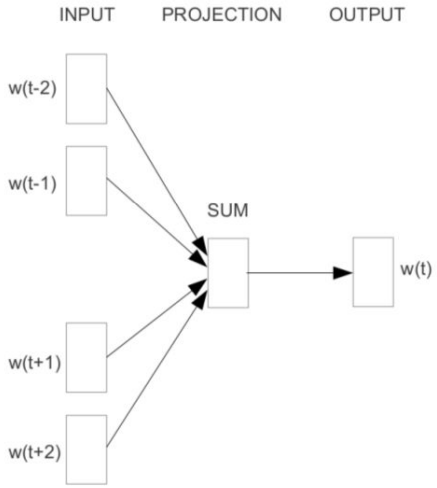
最后我们首先从中心词的左半部分找一条权重值最大的从根节点到叶子的路径，然后在这条路径所对应的句子集合中找出在右半部分权值最大的句子作为整篇文章的摘要。

3.Word Embedding（Word2Vec）

要将语言代入计算必须将其数字化，这就是编码的过程。然而，最基本的 1-N 编码实在是太稀疏了并且没有体现单词与单词之间的语义关系，因此我们需要一个比较小的空间将单词表示得更加密集些。Word Embedding 主要就是一个将不可计算的词转变为可计算的向量的过程。

在 Word Embedding 领域中最经典就是 Word2Vec 算法，这里只介绍 Word2Vec 中的 CBOW，该算法借助人工神经网络，其模型如下。首先选取当前词的前 N 个词和后 N 个词（N 为人为指定，图中的值为 2），通过一个或者几个全连接层进行投影变换，最后通过一个 Softmax 层进行分类输出，值得指出的是，输出层的第 K 个神经元的意义是一个条件概率 P(X[i]=K|X[i-N],X[i-N+1]...X[i-1],X[i+1]...X[i+N-1],X[i+N])，我们利用误差回传来完成对这个网络的训练，隐含层的词向量在每一次迭代的过程中都会更新。这样，词向量就充分体

现了该词的上下文语义信息并且维度可以人为指定（通过指定隐含层的神经元个数）。本质上讲，Word2Vec 得到的词向量也是基于上下文共现的（与 LDA 相同）。



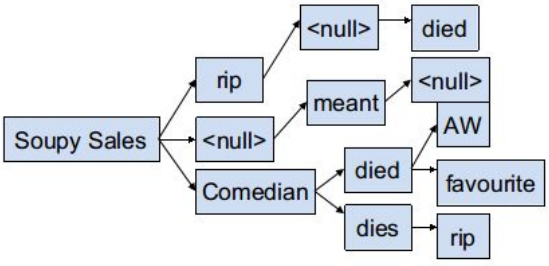
通过 Word2Vec 算法得出的词向量有一些很好的性质，其中最突出的就是意义相近的单词所对应的词项量的夹角比较接近；此外，词向量中包含了很多的隐式语义，例如 Vec(国王)-Vec(男性)=Vec(王后)-Vec(女性)。

4.对模型的相关改进

4.1 双向语法树

我们是针对第 2 部分提出的短文本摘要生成算法（下面称为原始算法）进行改进。首先，我们注意到的是原始算法是首先考虑中心词左侧的权值最大路径然后再考虑中心词右侧的。换句话说，也就是中心词右侧的部分只有其对应的句子左侧的权值最大的时候才会被考虑，这一点是不合理的。同样，我们先考虑中心词右侧的词也会出现类似的问题。针对原始算法的这个缺陷我提出一种双向的算法，即不再像原始算法一样构建一左一右两个语义树，而是仅仅构建一个语义树，该语义树的构建充分考虑了中心词左右两个方向附近的词，其从根节点往下依次是左侧 1 个词、右侧 1 个词、左侧 2 个词、右侧 2 个词.....同样取第 2 部分中的六个句子，由他们构建的语义树如下（仅仅画出了 4 层），其中<null>表示内容为空的节点。

Aw, Comedian Soupy Sales died.
RIP Comedian Soupy Sales dies at age 83.
My favorite comedian Soupy Sales died.
RT @NY: RIP Comedian Soupy Sales dies at age 83.
RIP: Soupy Sales Died Today.
Soupy Sales meant silliness and laughs.



在这样构建的语义树下，我们只需要找到从根节点到叶子节点权重最大的路径即可。

4.2 考虑句子的长短

对于一个语料库的摘要，它的长度应该适中，过长的摘要会夹杂很多冗余信息而过短的摘要会遗漏很多重点。原始算法没有考虑到摘要长短的问题，由于语义树中的各个节点不太可能为负数，所以原始算法偏向于提取长度比较长的句子，我们需要对比较长的句子给予一定惩罚。我们假设 K 为摘要句子的最佳值。对于语料集中的每一个句子 S，我们采用如下公式来计算其作为摘要的合理性。其中 Sum(S)表示 S 在语义树上对应的路径的权重和（对于双向的语义树，<null>节点的权重恒定为 0），Length(S)表示 S 的长度。

Weight(S)=Sum(S)/max{Length(S),K}

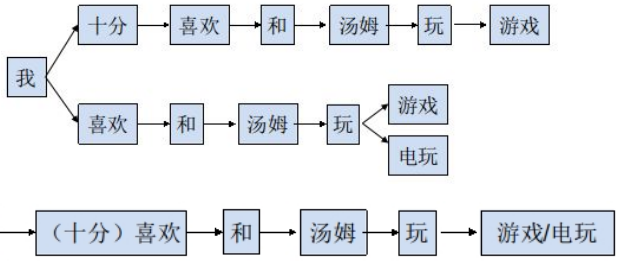
4.3 引入主题模型

从 4.1 部分构建的语义树可以看出 died 和 dies 两个相似的词应该合并，合并可以使语义树能够更加充分地体现语料集的中心思想（类似于非监督学习中的聚类），所以在文档的预处理阶段，我们需要做 Stemmer 和 Lemmatization。

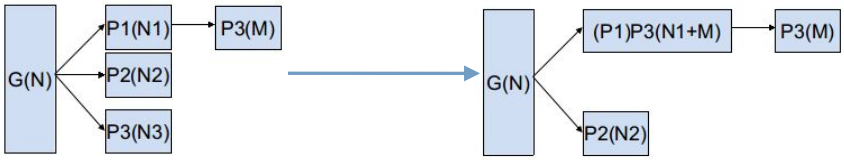
此外，不同于原始算法所针对的英文，本实验所针对的中文微博在语言上更加灵活。语料库出现很多分词系统无法识别的网络用语，加上中文庞大的词汇量和经常使用的修饰语，所有的这些都会使原始算法中的语义树变得又大又深，换句话说，也就是对整个语料库的聚类效果不是很好。针对中文文本，我们需要做的是将词语首先进行合并操作，合并后的词语可以出现在语义树的同一个节点；此外我们还要修饰部分进行剔除或者压缩，使得内容相似的词语得以出现在语义树的同一层方便合并。

举个例子，以下意义相近的三个句子按照原始算法就会分在三个不同的路径上，但是如果按照上述过程处理下就会分布在同一个路径上。

我喜欢和汤姆玩游戏
我喜欢和汤姆玩电玩
我十分喜欢和汤姆玩游戏



在本实验中，我们综合应用 LDA 和 Word2Vec 算法来寻找需要合并的词。我针对 LDA 的聚类结果中每一类中的词，选取他们夹角小于阈值的词对，如果该词对在语义树上比较接近（存在父子或者兄弟节点关系），则我们讲这两个词合并。至于挖掘可以忽略的修饰词，我们采用一种完全基于树的方法。如果一个节点所对应的句子数与其父节点的比值大于一个阈值并且其父节点存在一个与其对应词相同的兄弟节点，我们则认为这个节点的父节点是一个修饰词对应的节点。用图形表示如下所示，每个节点表示为 W(N)的形式，其中 W 表示单词，N 表示该节点对应句子的数量。如果 M/N1>K(阈值)，则我们认为 P1 为修饰词，可忽略,整个树的结果变成右侧。



在主题模型引入之后，原来语法树的节点数会变少，在这样的情况下，不论我们使用的是单向还是双向的语义树，所得到的摘要的信息量会更大并且更加紧凑。

5.实验

5.1 考虑摘要的长度

以雅安地震的语料集（约 2 万）为例，原始算法得到的得分最高的句子为（进行过分词处理，/后表示词性）：

四川/n 雅安/n 7/v 级/q 地震/n 08 年/m 汶川/n 地震/n 还/d 存在/v 很/d 真实/a 很/d 痛/a 的/u 记忆/n 13 年/m 雅安地震/n 虽然/c 身/n 在/p 内江/n 但是/c 比/p 那个/r 时候/n 还/d 要/v 害怕/v 担心/v 从/p 早上/t 考/v 完/v 试/v 源源不断/v 的/u 消息/n 问/v 我/r 还/d 好/a 吗/y 家里/s 怎么样/r 了/y 到/v 现在/t 看/v 到/v 那些/r 不/d 断/v 刷新/v 的/u 灾区/n 的/u 消息/n 忍不住/v 哭/v 了/u 愿/v 逝/v 者/k 安息/v 灾难/n 总/d 会/v 过去/t everything/x gonna/x be/x alright/x ./w

考虑到句子的长度并且归一化之后得到的得分最高的句子如下，默认的最佳摘要长度为语料集中所有句子长度的平均值。

四川/n 雅安/n 7/v 级/q 地震/n 我/r 大/a 雅安/n 必须/d 加油/v 啊/y ！/w ！/w ！/w

从上述结果可以明显看出原始算法有很大的冗余，但是考虑到摘要的长度归一化之后得到的结果要简练很多。

5.2 双向 vs 单向

我们对比单向语义树和双向语义树算法得分最高的几个句子，不论是单向还是双向语义树，我们均考虑句子长度的影响，并且设定 K 的值为 10.

单向语义树的结果：

大型/b 设备/n 无法/un 作业/v 四川/n 雅安地震/n 雅安/n 7/v 级/q 地震/n
刚才/t 是不是/v 又/d 震/v 了/u 雅安/n 7/v 级/q 地震/n
又/d 震/v 了/u 雅安/n 7/v 级/q 地震/n
雅安/n 7/v 级/q 地震/n 我/r 被/p 吓到/un 了/y …/w …/w
雅安/n 7/v 级/q 地震/n 雅安/n 2013/m 。/w
北京/n CCTV/x 记者/n 赶往/v 灾区/n 雅安/n 7/v 级/q 地震/n
雅安/n 7/v 级/q 地震/n 没事/v 的/u 只是/c 余震/n 有点/c 频繁/a
祈福/v 家乡/n 人们/n 能/v 坚持/v 抗震救灾/v 雅安/n 7/v 级/q 地震/n
雅安/n 7/v 级/q 地震/n 但愿/v 一切/r 都/d 快快/d 好/a 起来/v
雅安/n 7/v 级/q 地震/n 会/v 没/v 事/n 的/u

双向语义树的结果：

雅安/n 7/v 级/q 地震/n 我/r 被/p 吓到/un 了/y …/w …/w
雅安/n 7/v 级/q 地震/n 雅安/n 2013/m 。/w
雅安/n 7/v 级/q 地震/n 没事/v 的/u 只是/c 余震/n 有点/c 频繁/a
雅安/n 7/v 级/q 地震/n 但愿/v 一切/r 都/d 快快/d 好/a 起来/v
雅安/n 7/v 级/q 地震/n 会/v 没/v 事/n 的/u
雅安/n 7/v 级/q 地震/n 来/v 看看/v 这个/r 话题/n
雅安/n 7/v 级/q 地震/n 2013/m 4/n 20/m 8：02/m
雅安/n 7/v 级/q 地震/n 灾难/n 如此/r 之/u 近/a 我/r 在/p
雅安/n 7/v 级/q 地震/n 唉/e 四川/n 多灾多难/v 啊/y ！/w
雅安/n 7/v 级/q 地震/n 天亮/v 了/u 我/r 在/p

从上面的结果可以看出，双向语义树中前 10 名的句子每个句子都包括“雅安 7 级地震”字样且出现在句首，但是单向语义树则包括更多的冗余信息和低频词，从这点来说双向语义树相较于单向语义树有所改进。

5.3 修饰词和同义词挖掘

我们使用规模更大的关于钓鱼岛的语料集（约 44.7 万）来进行同义词和修饰词挖掘。

修饰语的挖掘结果在文件./doc/diaoyudao/classbased/phrase.txt 中（见附录）。下面是该文件最初几行的内容。

国/n 中美/n -> 中美国/n
姑夫/n 仙/n -> 仙姑夫/n
作品/n 仙姑夫/n -> 仙姑夫作品/n
买/v 拒/v -> 拒买/v
日货/n 拒买/v -> 拒买日货/vn
钓鱼岛/n 购买/v -> 购买钓鱼岛/vn
真相/n 幕后/s -> 幕后真相/sn
自/p 分享/v -> 分享自/vp
日货/n 抵制/v -> 抵制日货/vn
示威/v 游行/v -> 游行示威/v
活动/v 游行示威/v -> 游行示威活动/v
发酵/v 持续/v -> 持续发酵/v

同义词的挖掘结果在文件./doc/diaoyudao/classbased/vec-based-syn.list 中（见附录），该文件中每一行的记录结果为“<单词 1> <单词 2> <相似度（vec 夹角 cos 值）> <共现次数> <是否为同义词（1 为不是，0 为是）>”（由于修饰词的存在，会导致语义树中的词经常出现错位的现象，该现象会导致纯粹借助语义树挖掘同义词会出现错误，我们通过词向量的夹角来矫正，夹角太大的会被滤去，及最后一项的值标记为为 1）。下面列举了前几个最后一个标签为 0 的例子。

自古/d	*自古以来/d*	*0.6657135348803572*	*540*	*0*
咱们/r	*咱/r*	*0.6828401920410543*	*177*	*0*
宿营/v	*事/n*	*0.6550053998357239*	*157*	*0*
祝福/v	*祝/v*	*0.6689969001355657*	*146*	*0*
但/c	*但是/c*	*0.7039807428387965*	*136*	*0*
苍/n	*苍/a*	*0.8079926569394582*	*136*	*0*
它们/r	*他们/r*	*0.6837170814959247*	*114*	*0*
高/a	*在线/v*	*0.7193647631993548*	*112*	*0*
那么/r	*这么/r*	*0.7219259021116287*	*102*	*0*
能耐/n	*本事/n*	*0.8047163491522753*	*92*	*0*
一样/u	*以前/f*	*0.6859314037879471*	*85*	*0*
登广告/n	*广告/n*	*0.6782564829459515*	*74*	*0*

我们可以看出不论是同义词还是修饰词，大多数挖掘结果都是正确的，把同义词和修饰词引入到原始算法中，必然会减小原来的语义树的规模，对后续的摘要生成是有益的。

6.总结

本次 SRT 从研究 LDA 及其相关变种的论文开始，初步了解了主题模型在自然语言处理中的应用，综合实验室的资源和往届师兄师姐研究的问题，我最终确定将微博文本挖掘作为

本 SRT 的主要研究对象。在明确对象之后，我调研了很多针对短文本的摘要生成模型，最终决定使用基于语义树的模型进行改进。为了更好地将语言中的单词代入计算中，我学习了 Word Embedding 的相关知识，最终确定了“单向语义树->双向语义树->基于 Topic Model 的语义挖掘”的框架并进行实验。

在本次 SRT 进行的过程中，包括基本知识的学习、相关文献的阅读和代码的编写调试，我掌握了科学研究的基本技能。特别是最难的 LDA 部分，我首先阅读了布雷等人最早的文章，然后从引用这篇文章的较好文章中学习 LDA 的各个变种，这种从基本到衍生的阅读方法在实验室科研探究中十分常用。

作为本科生进入实验室的第一次尝试，由于时间的仓促和个人经验的缺乏，本次实验还有很多不足之处，最主要体现在两点：没有对挖掘的结果进行定量分析和提出的方法大多基于经验而非严谨的数学推导。不过，通过一个多学期的学习和探究，坚定了我对人工智能特别是自然语言处理的热爱，对于本次实验中用到的比较复杂的概率统计和机器学习知识（如神经网络），我会在之后的学习中继续研究。

最后感谢老师的指导和监督，实验室中师兄的帮助，是你们让我这个门外汉走进人工智能领域的大门。

7.参考文献

[1].David M. Blei, Andrew Y. Ng and Michael I. Jordan. 2003, Latent Dirichlet Allocation. Journal of Machine Learning Research, 3:993-1022.

[2].David Andrzejewski and Xiaojin Zhu. 2009, Latent Dirichlet Allocation with topic in set knowledge. Proceedings of NAACL.

[3].David Andrzejewski, Xiaojin Zhu, Mark Graven and Benjamin Recht. 2011, A Framework for Incorporating General Domain Knowledge into Latent Dirichlet Allocation using First-Order Logic. IJCAI 2011.

[4].Beaux Sharifi, Mark-Anthony Hutton and Jugal K. Kalita. 2010, Experiments in Microblog Summarization. IEEE International Conference on Social Computing.

[5].Jeffrey Nichols, Jalal Mahmud, Clemens Drews. 2012, Summarizing Sport Events Using Twitter. Proceedings of the 2012 ACM international conference on Intelligent User Interfaces.

[6].Charu C. Aggarwal and Chengxiang Zhai. Mining Text Data. 2012, Springer Science and Business Media.

附录：

I.实验代码结构

实验是在实验室开源框架的基础上改写的，*斜线部分*为框架代码

|-bin　　java 字节码
|-doc
　　|-diaoyudao
　　　　|-weibo.htm　　原始微博文档
　　　　|-input.txt　　预处理的输入文档
　　　　|-stopwords.dat　　停用词表
　　　　|-splited.txt　　进行分词之后的文档
　　　　|-selected.txt　　去长尾之后的文档
　　　　|-offstopword.txt　　去除停用词后的文档
　　　　|-offspeechtag.txt　　去除词性标签后的文档
　　　　|-speechchosen.txt　　选取特定的词性后的文档
　　　　|-speechchosenoffspeechtag.txt　　上一个文档去除词性标签
　　　　|-naive
　　　　　　|-single.tree　　单向语义树
　　　　　　|-single.summary　　单向语义树生成的摘要
　　　　　　|-double.summary　　双向语义树生成的摘要
　　　　|-classbased
　　　　　　|-phrase.txt　　挖掘的词组列表
　　　　　　|-vec-based-syn.list　　挖掘的同义词列表
　　|-yaan　　结构同./doc/diaoyudao
|-lib　　库
|-src　　java 源代码
　　|-ICTCLAS/I3S/AC　　中科院分词软件调用代码
　　|-*db*　　*与数据库的连接部分*
　　|-*demo*　　*demo*
　　|-LDAVariety　　LDA 基类
　　|-*DFLDA*|-*InfiniteLDA*|-*newHLDA*　　*三种 LDA 变种*
　　|-*DirichletForest*　　*狄利克雷森林*
　　|-util　　一些基本类
　　|-*Fpgrowth*|-*NodeClass*|-*org*|-*sentiment*|-*thu*|-*TM*　　*框架中与实验无关的部分*
　　|-preprocess　　预处理部分
　　　　|-GenDoc.java　　除去 htm 源文本的标签
　　　　|-PreProcess.java　　预处理程序
　　　　|-Config.java　　包内全局变量
　　|-tf_idf　　文献[2]Hybrid TFIDF 算法
　　|-naiveSummary　　非基于主题的摘要生成

|-Node.java	语义树中的节点
|-Summary.java	单向语义树生成算法
|-BothWaySummary.java	双向语义树生成算法
|-Config.java	包内全局变量
|-Lib.java	包内辅助类
|-classbBasedSummary	基于主题的摘要生成
|-Node.java	语义树中的节点
|-VectorLoader.java	加载词向量
|-SynPair.java	提取同义词
|-LinkBasedOnVec.java|-PhraseFinder.java	提取固定词组
|-Config.java	包内全局变量
|-Lib.java	包内辅助类
|-TopicModelData	主题模型数据
|-word2vec	word2vec 库

代码的运行平台为 Windows，Java 版本需要 Java 7 以上。

II.外部链接

Github 上的源代码：https://github.com/liuchen11/WeiboSummarization