不过在输入层添加了文档向量的学习. 如果两篇文档具有较多相似的词语, 那它们的文档向量也是比较接近的. 实验验证了文档表示在分类任务上有不错的效果. 现有的文档表示模型没有考虑文档中词语的重要性, 即使两篇文章有较多相似的但不重要单词, 也不能认为两篇文档相似, 所以在学习文档表示时, 考虑单词重要性尤为关键. 因此本文在表示文档的时候着重考虑了各部分的重要性.

1.2 注意力机制

注意力机制首先应用在图像问题中[8,16], 该研究动机来源于人类的注意力机制, 在图像和自然语言处理问题中, 可以看成图像或者文本中不同部分的重要性体现.

Bahdanau 第一次在机器翻译模型中引入了注意力机制[9], 之后注意力机制在自然语言处理中得到广泛应用. Wang 尝试把注意力机制引入到无监督句子表示学习上, 扩展 PV-DM 方法, 提出了 aCSE 模型[14], Wang 认为窗口中的所有单词重要性是不一样的, 上下文的每个单词应该赋予一个权值, 这个权值依赖距离目标词的位置, 并且在训练的时候得到. 这种注意力的构造是一种局部的注意力, 体现的是局部重要性, 仍然无法看出一个词对于理解整个句子或文档的全局重要性.

Yang 为了解决文档分类问题, 提出了一个层级注意力模型 (Hierachical Attention Network, HAN)[10]. 该文考虑一篇文档具有的层级结构, 即文档由句子构成, 句子由词构成, 在构建文档的表示之前先构建句子表示, 然后通过句子表示得到文档的最终表示, 单词和句子的重要性在不同的文档中都可能不同. Yang 构建的层级注意力模型是一个监督的学习模型, 并且仅限于应用在单个自然语言处理任务上.

在基于无监督学习的文档表示模型中, 现有的模型没有考虑文档的层级关系, aCSE[14]只考虑单词的局部重要性, 而且这种基于位置的注意力不合理, 没有考虑单词与单词的关系, 它适用于较短的文本. 另一方面, 在基于监督学习的文档表示模型中, HAN 虽然考虑了文档的层级关系和使用了层级注意力机制, 但由于监督学习局限性, 无法处理大量的未标记的文本, 学习的特征受到局限.

结合上述的模型, 本文把大量的未标记文本数据利用起来, 同时考虑到文档的层级结构和文本的注意力机制, 通过无监督的学习方式得到文档的表示, 称为基于层级注意力机制的无监督文档表示学习方法. 该模型可以高效的学习海量数据特征, 通过浅层模型得到文本的语义表示.

2 HADR 模型

2.1 CBOW 模型和 Skip-Gram 模型

词嵌入模型一般通过大量的无监督文本训练词向量. 在词嵌入模型中, 假设词汇表是V, 一般的, 中文文本需要将句子进行分词操作才能统计词汇表, 每个词将表示成一个长度为d的向量, 所有的词向量可以组成一个词矩阵$\mathrm{W}\in R^{d\times|\mathrm{V}|}$, 词$w_i,\{i=1,2,\cdots,|\mathrm{V}|\}$的向量表示可以写成$v(w_i)$. 给定一篇文档可以表示成$S=\{w_1,w_2,\cdots,w_l\}$, l是文档的长度. 大部分词嵌入模型都需要构建句子中的滑动窗口, 假设w_t为目标单词, w_t的上下文由相邻的一些单词构成, 表示为$c_t=\{w_{t-k},\cdots,w_{t+k}\}$, 不包括$w_t$, c_t可以看成一个随着t变化而移动的滑动窗口, $2k$是窗口的大小. Word2Vec 的两个模型可以写成极大化如下目标函数的形式:

$$L(S)_{\mathrm{CBOW}}=\frac{1}{l-2k-2}\sum_{t=k+1}^{l-k}\log P(w_t|c_t) \tag{1}$$

$$L(S)_{\mathrm{Skip-gram}}=\frac{1}{l-2k}\sum_{t=k+1}^{l-k}\sum_{-k\leqslant i\leqslant k,i\neq 0}\log P(w_{t+i}|w_t)$$

其中, c_t是w_t的上下文, 概率函数可以表示成一个 softmax 函数:

$$P(w_t|c_t)=\frac{e^{y_{w_t}}}{\sum\limits_{w_i\in V}e^{y_{w_i}}} \tag{2}$$

在 CBOW 模型中, y_{w_t}表示成隐藏变量和w_t向量的内积$y_{w_t}=h(w_t)v(w_t)^{\mathrm{T}}$, 在 Skip-Gram 模型中, $y_{w_t}=v(w_{t+i})v(w_t)^{\mathrm{T}}$. CBOW 中的隐藏变量可以用上下文的窗口向量表示, 一般是窗口内所有单词向量的均值或者相连, 而 Skip-Gram 中的隐藏变量就是w_t上下文中的一个单词向量. 如果是利用均值表示, 隐藏变量写成如下形式:

$$h(w_t)=v(c_t)=\frac{1}{2k}\sum_{-k\leqslant i\leqslant k,i\neq 0}v(w_{t+i}) \tag{3}$$

训练过程需要大量的文本语料库, 将语料库构造一系列的滑动窗口, 利用随机梯度下降和反向传播算法优化, 不断的对参数更新, 同时对词向量进行更新, 得到最终的语言模型. CBOW 和 Skip-Gram 的优势在

于用一个比较简单的神经网络模型就可以构造出语言模型, 同时得到了具有语义相关性的单词分布式表示. 而且论文利用负采样 (Negative Sampling), 子采样 (Subsampling) 和层级 softmax 等技术进一步提升了模型的效率[3,4].

2.2 PV-DM 和 PV-DBOW

PV-DM 和 PV-DBOW 类比 Word2Vec 的方式学习文档的特征表示, 分别在 CBOW 和 Skip-Gram 模型中添加一个段 ID (Paragraph ID), 这个段 ID 就是指句子或者文档的表示向量, 维度与词向量相同, 记为$v(S)$.

基于 CBOW 的句子表示模型称为 PV-DM, 它在构造隐藏变量时联合窗口内的词向量和文档向量$v(S)$, 隐藏向量可以写成如下形式:

$$h(w_t)=\frac{1}{2k}\left(\sum_{-k\leqslant i\leqslant k,i\neq 0} v(w_{t+i})\right)+v(S) \tag{4}$$

PV-DBOW 是直接通过当前的段 ID 来预测文档中所有的目标单词. 隐藏向量就是文档向量$h(w_t)=v(S)$. 相比 Word2Vec 模型, PV-DM 和 PV-DBOW 的优化方式相似, 每篇文档多出一个文档向量的更新, 它们的结构如图 1 所示.

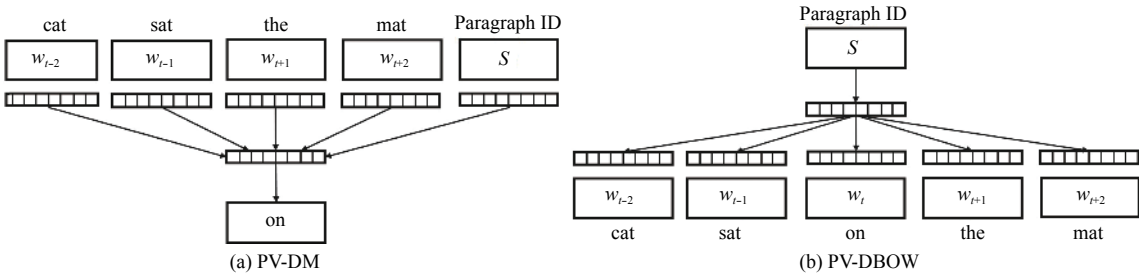

图 1 PV-DM 和 PV-DBOW 模型

2.3 HADR 模型

HADR 模型是本文提出的一个基于注意力机制文档表示模型, 该模型研究了文档的层级结构, 提出一种基于层级的注意力机制利用在文档的表示学习中. 相比于 PV-DM 和 aCSE 来说考虑更加词与词之间的相关性[7,14].

假设文档具有层级关系, 文档由句子构成, 句子由单词构成. $D=\{S_1,S_2,\cdots,S_N\}$, N 表示文档包含的句子的个数, 同样的, 第 n 个句子可以表示成$S_n=\{w_{n,1},w_{n,2},\cdots,w_{n,l_n}\}$, l_n是第 n 个句子的长度. 与 Word2Vec 结构类似, 假设目标单词是$w_{n,t}$, 上下文可以表示成$c_{n,t}=\{w_{n,t-k},\cdots,w_{n,t+k}\}$. 为了同时得到句子向量和文档向量, 层级结构语言模型通过句子, 文档和窗口单词来预测目标单词, HADR 模型的目标函数如下:

$$\mathrm{L}(D)=\frac{1}{(l_n-2k)*N}\sum_{n=1,2,\cdots,N}\sum_{t=k+1}^{l_n-k}\log P(w_{n,t}|c_{n,t},S_n,D) \tag{5}$$

通过窗口向量$v(c_{n,t})$, 句子向量$v(S_n)$, 文档向量$v(D)$三个向量构造当前窗口的隐藏变量, 然后通过隐藏变量和单词向量构造的 softmax 函数实现目标单词$w_{n,t}$的预测. 最关键的问题就是如何构建这个隐藏向量才能体现文档中的重要组成部分. 下面介绍通过层级的方式来构造层级的隐藏变量, 构造一种层级的注意力机制.

相比于 CBOW 模型, 为了体现窗口中单词对窗口向量的贡献不一样, 在将所有词向量相加的时候, 给每个词向量赋予一个权值. 表示成如下形式, a^0代表 0 级注意力机制, 窗口向量也可以看成 0 级隐藏变量:

$$h^0(w_{n,t})=v(c_{n,t})=\sum_{-k\leqslant i\leqslant k,i\neq 0} a^0_{n,t+i}(w_{n,t})v(w_{n,t+i}) \tag{6}$$

在上下文中, 窗口内的词与目标词语义越相近, 它对窗口的贡献越大, 赋予更大的权值. 例如窗口单词是“the cat is”, 目标单词是“playing”, “cat”与“playing”相关, 权值越大, “the”、“is”与“playing”不相关, 权值小. 由于词向量的语义相关性, 可以通过向量的内积来表示单词之间的相关性. 0 级注意力可以通过归一化的向量内积表示:

$$a^0_{n,t+i}(w_{n,t})=\frac{\exp(v(w_{n,t+i})v(w_{n,t})^{\mathrm{T}})}{\sum_{-k\leqslant c\leqslant k,c\neq 0}\exp(v(w_{n,t+c})v(w_{n,t})^{\mathrm{T}})}$$

0 级注意力机制的设计是针对一个单词的局部注

意力. 另外一方面, 本文希望能得到一个句子或者一篇文档的全局注意力, 也就是针对句子或者文档中的每个单词的重要性. 当词向量和句子向量比较接近时, 模型给句子赋予更大的权值, 这样 1 级注意力就可以用加入 sigmoid 函数的词向量和句向量内积表示, 如下:

$$a_t^1 = \text{sigmoid}(v(w_{n,t})v(S)^{\mathrm{T}})$$

Sigmoid 函数作为神经网络的激活函数, 形式如下 $\text{sigmoid}(x) = 1/(1+\exp(-x))$. 与 PV-DM 模型类似, 1 级隐藏变量就可以表示成上一级隐藏向量和句子向量的加权求和:

$$h^1(w_{n,t}) = h^0(w_{n,t}) + a_{n,t}^1 v(S_n)$$

a_t^1表示成句子中的目标词的重要性, n 表示窗口所属第 n 个句子, t 表示为句子中的第 t 个词. 相比公式 (4) 中 PV-DM 句子向量的权值为 1, HADR 模型使用了一个句子和词之间关系的权值, 也就是 1 级注意力, 将上一级的隐藏向量和句子向量组合在一起. 与 1 级隐藏向量构造方式相同, 2 级隐藏变量利用 1 级隐藏向量和文档向量加权求和方式得到, 权值和隐藏向量的计算分别如下:

$$a_n^2 = \text{sigmoid}(v(S_n)v(D)^{\mathrm{T}})$$

$$h^2(w_{n,t}) = h^1(w_{n,t}) + a_n^2 v(D)$$

至此, 最终的隐藏向量构造出来了, a_n^2表示句子 S_n在文档 D 中的重要性. 接下来的步骤与 PV-DM 相同了, 通过最终隐藏向量去预测目标词$w_{n,t}$. HADR 模型在更新文档向量、句子向量和词向量的同时更新各级的注意力值, 并且所有的变量直到得到最优的模型更新停止, 最终得到具有更强语义的句子表示和文档表示, 并且量化了词在句子中的重要性 (注意力) 和句子在文档中的重要性. HADR 结构如图 2 所示.

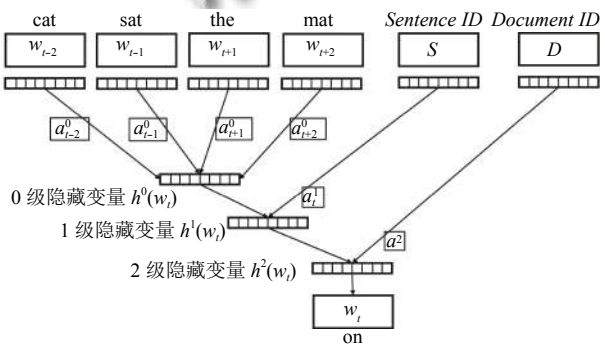

图 2 层级注意力结构的文档表示模型

3 实验与结果分析

为了研究层级注意力机制在文档表示中的作用, 本文从情感分析实验与现有的未加入注意力机制的 PV-DM 和 Word2Vec 模型进行对比. 在这个章节中, 本文开始介绍模型使用的数据集, 然后介绍实验代码的实现以及参数的设置, 最后介绍基于文档表示的情感分析, 并且与现有一些模型对比.

3.1 IMDB 数据集和预处理

IMDB (Internet Movie Data Base) 是英文的电影评分数据集, 每条评论包括一条文本, 可能是一个句子 (sentence), 也可能是由多个句子组成的文档 (document). 在所有 IMDB 数据中, 一部分评论已经打分为 1-10, 更高的评分表示用户更加喜欢该电影, 对应的评论也具有更积极地评价. 更多的, 把打分划分为消极的 (1, 2, 3, 4 分)、积极的 (7, 8, 9, 10 分), 将中性打分 (5, 6 分) 的评论数据删除. IMDB 情感分类的任务就是给定一条评论文本, 预测它的情感是积极的还是消极的. 除了已打分的评论, 还有一部分评论没有任何打分, 本文的模型通过无监督学习方式把这些未标记评论也加入到单词向量的学习中来. 图 3 中展示了 100 000 条积极、消极和未标记评论数据的分布.

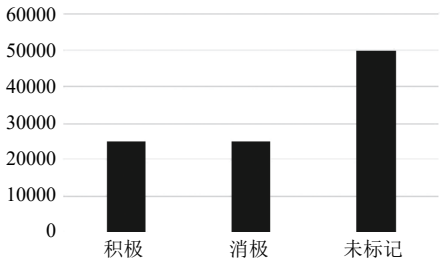

图 3 IMDB 积极、消极和未标记数据大小

在实验中, 本文对评论数据进行了预处理操作, 删除停用词 (stop words), 例如在英文中的停用词有: the, a, of 等. 在情感分析中, 停用词的作用非常小, 几乎不影响整个句子的含义. 同时将所有的字符都转化成小写字符, 删除语料库中频率低于 5 个的单词, 最终得到的词汇表大小为 64 720. 通过这样一系列的预处理操作降低了计算复杂度.

3.2 实现与参数设置

本文扩展 Python 库 gensim 中 Doc2Vec 脚本, 实现了文本提出的 HADR 模型. 为了让 HADR 模型和现有的模型具有可比性, 文本使用了相似的参数设置: 初

始的学习率 alpha 设置为 0.05, 滑动窗口大小为 5(目标单词左右各 5 个单词), 负采样大小为 25 个单词, 子采样取值为 10e–3. 训练的时候采用了分层 softmax, 删除了词频小于 5 的单词. 为了使得单词向量和句子向量充分的学习, HADR 算法和对比算法都迭代 20 次. 在对比模型的时候使用相同的向量维度进行对比.

3.3 情感分析实验

通过模型得到文档的特征表示之后, 本文使用 IMDB 数据情感分析来评价文本表示学习的性能. 电影评论信息能表示一个电影的评价, 它代表了一个电影的商业价值, 对电影评论进行情感分析具有重大的意义. 实验通过给定的训练集来预测测试集合中评论的情感分类, 分类器使用了来自 scikit-learn 库的逻辑回归代码, 逻辑回归是一个成熟的特征分类模型, 在很多分类问题上取得不错的效果. 本文使用 50 000 个已标记的数据进行情感分类实验, 利用 5 折交叉验证进行实验, 也就是将数据分成 5 份, 其中 4 份作为训练数据, 剩下的一份作为测试数据, 最终取 5 次实验的均值值作为指标.

本文对比了一些文本表示模型, 其中包括:

(1) Word2Vec[4,5]: Word2Vec 模型得到词向量, 参数设置和本文模型相似, 文档表示向量通过所有的词向量相加得到 (Google 的 C 代码);

(2) Doc2Vec[7]: 通过 gensim 中的 Doc2Vec 脚本得到的文档表示, 参数设置与本文的模型相似 (gensim 实现);

(3) TF-IDF: 通过 TF-IDF 算法统计文本的词频-逆文档频率作为的文本特征 (scikit-learn 实现).

本文使用相同参数的逻辑回归分类器对不同模型得到的文档表示进行分类实验, 实验通过正确率 (Accuracy) 来评价, 正确率越高表明模型的效果越好. 文本对比了在不同维度下, 不同模型的情感分类效果, 如图 4 所示. 我们可以看出随着维度的增长, 不同模型都呈现的性能都有提升的趋势, HADR 模型在考虑了文档中单词和句子的重要性之后取得了最好的效果, 并且在 dim=200 的时候效果基本接近最佳, 之后增大文本表示维度对情感分类的效果影响不大.

同时实验对比了 Doc2Vec 和 HADR 模型在不同迭代步数的分类正确率, 两个模型使用相同的向量维度 dim=200, 而且运行到 20 最大的迭代步数, 其他参数与 3.2 章节相同. Doc2Vec 和 HADR 模型实验对比结果如图 5 所示. 同样的每次分类使用 5 折交叉验证取 5 次实验的平均值. 从图中可以看出, 两个模型在随着迭代步数增长, 性能都有所提升, 而且迭代步数达到一定步数, 性能提升比较小.

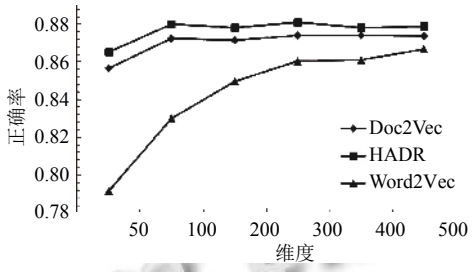

图 4 HADR 算法与对比算法的分类正确率对比

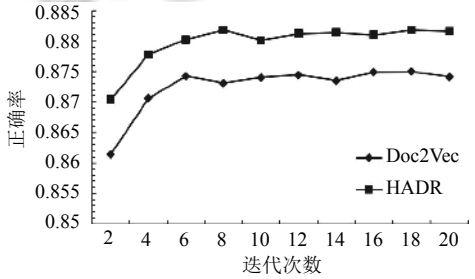

图 5 HADR 算法与 Doc2Vec 不同迭代次数的性能

4 结论与展望

基于 Le 等人提出的 PV-DM 算法[7], 本文在考虑了句子中不同单词具有不同重要性以及文档中句子也具有不同重要因素, 提出一种具有层级结构的注意力模型来学习文本表示. 改进的算法不仅得到了文档更好的表示, 也得到文档的多级表示, 同时通过这样无监督的注意力模型得到句子中每个单词重要性以及文档中每个句子的重要性, 这样将文本中每个部分的注意力值量化出来. 下一步工作将继续考虑文本的其他因素学习文本表示, 模拟人类遗忘机制, 人类在阅读一段文本的时候不仅会将注意力转移到几个关键的单词上或者句子上还会部分遗忘之前看的内容. 之后的工作希望能构建一个具有遗忘机制的文本表示模型.

参考文献

1 马胜蓝. 基于深度学习的文本检测算法在银行运维中应用. 计算机系统应用, 2017, 26(2): 184–188. [doi: 10.15888/j.cnki.csa.005628]

2 Posadas-Durán JP, Gómez-Adorno H, Sidorov G. Application of the distributed document representation in the authorship attribution task for small corpora. Soft

Computing, 2017, 21(3): 627–639. [doi: 10.1007/s00500-016-2446-x]

3 Bengio Y, Réjean D, Pascal V, *et al*. A neural probabilistic language model. Journal of Machine Learning Research, 2003: 1137–1155.

4 Mikolov T, Sutskever I, Chen K, *et al*. Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems, 2013: 3294–3302.

5 Mikolov T, Chen K, Corrado G, *et al*. Efficient estimation of word representations in vector space. arXiv preprint arXiv, 2013: 1301.3781.

6 Ryan K, Zhu YK, Salakhutdinov R, *et al*. Skip-thought vectors. Advances in Neural Information Processing Systems, 2015.

7 Le QV, Mikolov T. Distributed representations of sentences and documents. arXiv preprint arXiv, 2014: 1405.4053.

8 Kingma DP, Welling M. Auto-encoding variational Bayes. arXiv preprint arXiv, 2013: 1312.6114.

9 Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv, 2014: 1409.0473.

10 Yang ZC, Yang DY, Dyer C, *et al*. Hierarchical attention networks for document classification. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016. [doi: 10.18653/v1/N16-1174]

11 Pennington J, Socher R, Manning C. Glove: Global vectors for word representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014. 1532–1543. [doi: 10.3115/v1/D14-1162]

12 Levy O, Goldberg Y. Neural word embedding as implicit matrix factorization. Advances in Neural Information Processing Systems, 2014: 2177–2185.

13 Dai AM, Olah C, Le QV. Document embedding with paragraph vectors. arXiv preprint arXiv, 2015: 1507.07998.

14 Wang YS, Huang HY, Feng C, *et al*. Cse: Conceptual sentence embeddings based on attention model. The 54th Annual Meeting of the Association for Computational Linguistics. 2016. [doi: 10.18653/v1/P16-1048]

15 Hill F, Cho K, Korhonen A. Learning distributed representations of sentences from unlabelled data. arXiv preprint arXiv, 2016: 1602.03483.

16 Mnih V, Heess N, Graves A. Recurrent models of visual attention. Advances in Neural Information Processing Systems, 2014: 2204–2212.