A Hybrid Document Feature Extraction Method Using Latent Dirichlet Allocation and Word2Vec

NEW FRAMEWORK

LDA 可以用来描述文档之间的全局关系，Word2Vec 可以以非常局部的方式来预测单词。所以我们将两者合并，使用一种更加全面的向量来表示文档，同时，使用密度向量的新型表示方法增强了用于 NLP 任务的区分和预测能力。

如 Fig. 3(b) 所示，新方法将单词、文档和主题投影到一个高维语义空间中。一个**文档向量**被认为是一个单向量 (single vector)，是文档中所有单词的质心 (centroid)——正如 Word2Vec 在投影层做的那样。另外，由于每篇文档长度不一，因此文档向量需要除以该文档中的单词数，以保证度量标准的统一性。我们以一种相似的方法构造**主题向量**，但这会有一些复杂。我们使用每个主题下概率最高的 h 个单词来表示该主题，然后重新调整这些单词的概率，将其作为单词的权重。因此不同的单词对主题的贡献程度也不同。我们度量每篇文档和各个主题之间的欧氏距离，使得文档可以用距离分布来表示。

详细来讲,给定一个文档集合 $D = \{d_1, d_2, \ldots, d_n\}$,其词汇表构建自 N 个单词集合 $\{w_1, w_2, \ldots, w_N\}$. 通过训练 D，LDA 输出潜在主题 $\{t_1, t_2, \ldots, t_T\}$(即 GibbsLDA++ 输出的 theta 文件，doc-topic 分布) 和每个主题 t_i 中的单词概率 (即 GibbsLDA++ 输出的 phi 文件，topic-word 分布)，其中，主题 t_i 下的第 j 个单词表示为 θ_{i_j}. Word2Vec 训练 D，并将词汇表中的每个单词表示为一个固定长度的向量 $\{v(w_1), v(w_2), \ldots, v(w_N)\}$. 为了生成主题向量，从主题 t_i 中选择 h 个概率最高的单词。同时，使用 Eq.(1) 将主题 t_i 中的单词概率重新调整为权重。在 Eq.(2) 中，通过对每个单词的向量与其权重的乘积求和，即可求得**主题向量** $v(t_i)$.

$$\omega_i = \frac{\theta_i}{\sum_{n=1}^{h} \theta_n} \tag{1}$$

$$v(t_i) = \sum_{n=1}^{h} \omega_{i_n} v(w_{i_n}) \tag{2}$$

- Eq.(1) 批注：
  - θ_i 是主题 t_i 对应的单词向量，即 phi 文件中行号为 t_i 对应的那一行;
  - $\sum_{n=1}^{h} \theta_n$ 应该是写错了，应为 $\sum_{n=1}^{h} \theta_{i_n}$，即对主题 t_i 下概率最高的 h 个单词的概率求和.
- Eq.(2) 批注：

  - 求和过程中的 $\omega_{i_1}, \omega_{i_2}, \ldots, \omega_{i_h}$,是主题 t_i 下概率最高的 h 个单词的权重值;$w_{i_1}, w_{i_2}, \ldots, w_{i_h}$ 是具体的单词,将其作为索引可以从 word2vec 的训练结果中取出对应的词向量 $v(w_{i_n})$.

- Eq.(1) 和 Eq.(2) 中的向量维数都是 N，N 等于词汇表的大小。

接下来，计算通过 Eq.(3) 计算**文档向量** $v(d_i)$，其中 c 为文档 d_i 中的单词数。

$$v(d_i) = \frac{\sum_{n=1}^{c} v(w_{i_n})}{c} \tag{3}$$

因此，每篇文档可被表示为语义空间中该文档到各个主题的距离分布，该距离由 Eq.(4) 计算。

$$distance(v(d_i), v(t_j)) = |v(d_i) - v(t_j)| \tag{4}$$

where $i = 1, 2, \ldots, n; j = 1, 2, \ldots, T$