Don't Forget the Quantifiable Relationship between Words: Using Recurrent Neural Network for Short Text Topic Discovery

by Heng-Yang Lu, Lu-Yao Xie, Ning Kang, Chong-Jun Wang and Jun-Yuan Xie

—— 李家豪 151220050

topic model

There are countless data emerging everyday which contain valuable information to be mined. Particularly in recent years, the Internet has totally changed our life. For example, more and more people express their opinions through social network, and journalists are used to post their news on the Internet. We can hardly analyse these massive data directly, that's why we need a tool like topic model to help us organize and summarize digital data automatically.

Lots of work has been done in the research field of topic model. Early studies like probabilistic latent semantic analysis (PLSA) (Hofmann 1999) and latent dirichlet allocation (LDA) (Blei, Ng, and Jordan 2003) are two classic topic models widely used for discovering hidden topics from text corpus.

LDA(Latent Dirichlet Allocation)

文档中的每个单词都是以一定概率抽取主题，一定概率从主题中抽取的。

文档集合D = {d1,d2,...,dn}，d = <w1, w2, ...,wn>

主题集合T = {t1, t2, ..., tk}

对文档d，对应到主题的概率 θd<pt1,...,ptk>

对主题t，对应到单词的概率 ϕt<pw1,...,pwm>

核心公式：p(w|d)=p(w|t)*p(t|d)

LDA(Latent Dirichlet Allocation)

对文档ds中的第i个单词wi，令该单词对应的topic为tj。
核心公式变为pj(wi|ds)=p(wi|tj)*p(tj|ds)

枚举所有的topic，得到tj为所有情况时的pj(wi|ds)，选择一种决定wi的topic的方法(如选取使pj(wi|ds)最大时的topic。

选择后，θd和ϕt都会被影响和改变。进行n次迭代后，θd和ϕt便能收敛接近于我们想要的LDA模型。

PLSA(probabilistic latent semantic analysis)

p(w|d)=∑zp(w|z)p(z|d)

$$\sum_{k=1}^{Z} Q(z_k | d_i, w_j) \ln p(w_j | z_k) p(z_k | d_i)$$

其中Q是z的分布函数，表示在给定参数的情况下(w,d)，z的后验概率。

E-step :

$$Q(z_k | d_i, w_j) = \frac{p(w_j | z_k) p(z_k | d_i)}{\sum_{k=1}^{K} p(w_j | z_k) p(z_k | d_i)}$$

M-step:

$$E(n(d, w) \ln p(w | z) p(z | d)) = \sum_{i=1}^{N} \sum_{j=1}^{M} n(d_i, w_j) \sum_{k=1}^{Z} Q(z_k | d_i, w_j) \ln p(w_j | z_k) p(z_k | d_i)$$

However, posting short text data like tweets or online questions on the Internet is becoming popular, we have to deal with short text more often. Different from regular text data, the sparsity of short text content brings challenge to traditional topic models because words are too few to learn and analyze from original corpus.

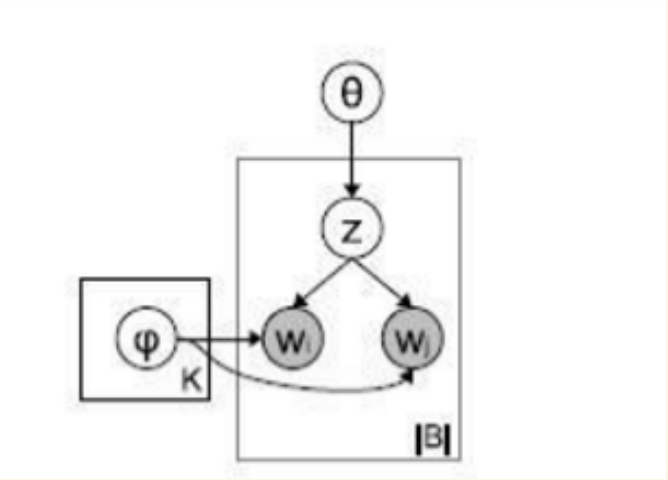
extending the original short texts into longer ones by aggregating similar texts:
aggregated texts which were posted by the same author before using LDA

bringing in related texts from online search results

Another creative idea alleviates the problem by constructing word pairs or word groups to represent the original texts.(BTM, WNTM, PMI)

BTM(biterm topic model)

1. For each topic z
   (a) draw a topic-specific word distribution $\phi_z \sim \mathrm{Dir}(\beta)$
2. Draw a topic distribution $\theta \sim \mathrm{Dir}(\alpha)$ for the whole collection
3. For each biterm b in the biterm set B
   (a) draw a topic assignment $z \sim \mathrm{Multi}(\theta)$
   (b) draw two words: $w_i, w_j \sim \mathrm{Mulit}(\phi_z)$



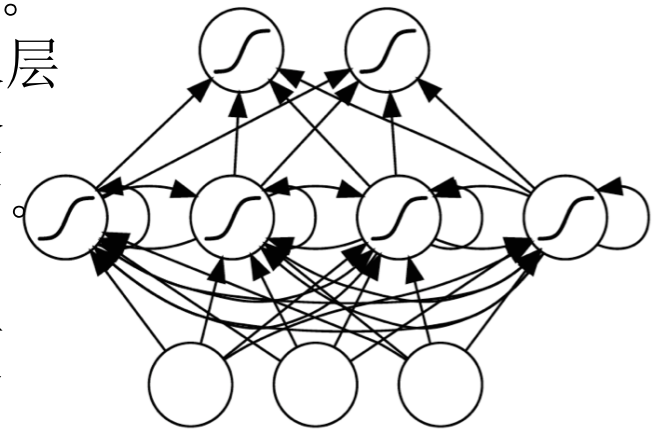
$$P(z|d) = \sum_b P(z|b)P(b|d).$$

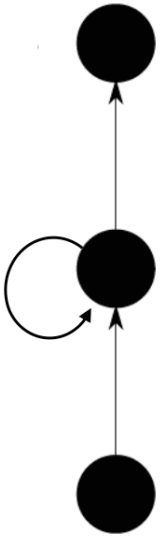
$$P(z|b) = \frac{P(z)P(w_i|z)P(w_j|z)}{\sum_z P(z)P(w_i|z)P(w_j|z)},$$

$$P(b|d) = \frac{n_d(b)}{\sum_b n_d(b)},$$

So we prefer to learn this relationship by training recurrent neural networks (RNN) not only relying on its learning skills but also on its intelligent memory. At the same time, to filter high-frequency words, we apply classic inverse document frequency (IDF) (Sparck Jones 1972) for each word. We call this model as RNN-IDF based Biterm Short-text Topic Model (RIBS-TM).

RNN(recurrent neural networks)





RNN的目的使用来处理序列数据。在传统的神经网络模型中，是从输入层隐含层再到输出层，层与层之间是全连接的，每层之间的节点是无连接的。但是这种普通的神经网络对于很多问题却无能无力。例如，你要预测句子的下一个单词是什么，一般需要用到前面的单词，因为一个句子中前后单词并不是独立的。RNN之所以称为循环神经网路，即一个序列当前的输出与前面的输出也有关。具体的表现形式为网络会对前面的信息进行记忆并应用于当前输出的计算中，即隐藏层之间的节点不再无连接而是有连接的，并且隐藏层的输入不仅包括输入层的输出还包括上一时刻隐藏层的输出。理论上，RNN能够对任何长度的序列数据进行处理

IDF(Inverse document frequency)　逆文本频率指数

我们很容易发现，如果一个关键词只在很少的网页中出现，我们通过它就容易锁定搜索目标，它的权重也就应该大。反之如果一个词在大量网页中出现，我们看到它仍然不很清楚要找什么内容，因此它应该小。概括地讲，假定一个关键词 w 在 Dw 个网页中出现过，那么 Dw 越大， w 的权重越小，反之亦然。

log(D/Dw)　　其中 D 是全部网页数

RIBS-TM

1. Learn prior knowledge β from corpus D.
2. Draw $\theta \sim$ Dirichlet (α).
3. For each topic $k \in [1, K]$
   (a) draw $\phi_{k,w_i} \sim$ Dirichlet (β_i).
   (b) draw $\phi_{k,w_j} \sim$ Dirichlet (β_j).
4. For each biterm $b \in \mathbf{B}$, where $b = (w_i, w_j)$
   (a) draw $z \sim$ Multinomial (θ).
   (b) draw $w_i \sim$ Multinomial $(\phi_{z_{w_i}})$.
   draw $w_j \sim$ Multinomial $(\phi_{z_{w_j}})$.

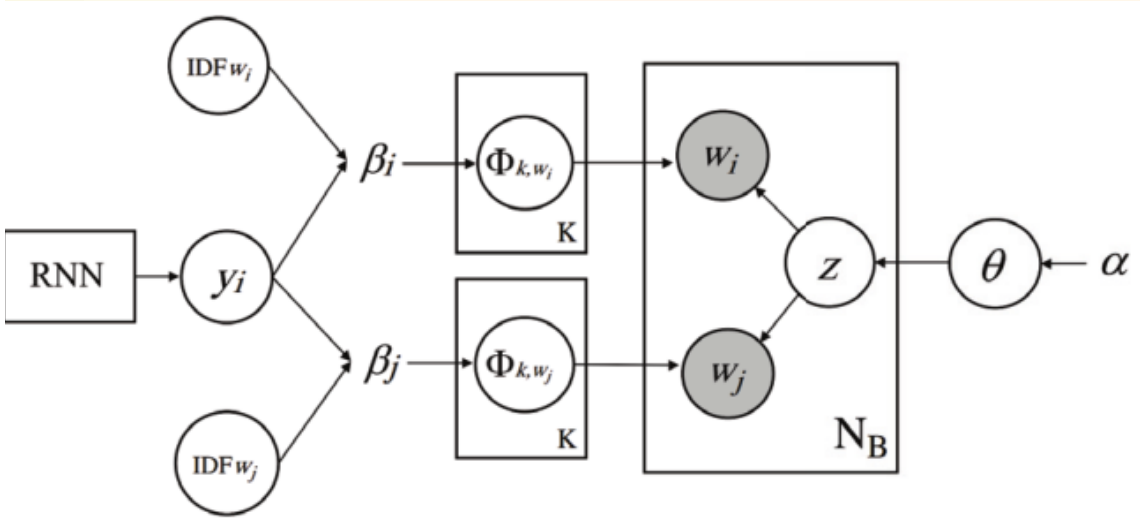


Figure 2: Graphical representation of RIBS-TM

Prior Knowledge Learning

- If two words are more likely to appear in the same generative sentence, they are more related.
- If two words are far away from each other in the same sentence, the relationship between them shall be weakened.

The input layer $x_t \in \mathbb{R}^{L+H}$ is defined as $x_t = [w_t, h_{t-1}]$, we can compute hidden and output layers with x_t:

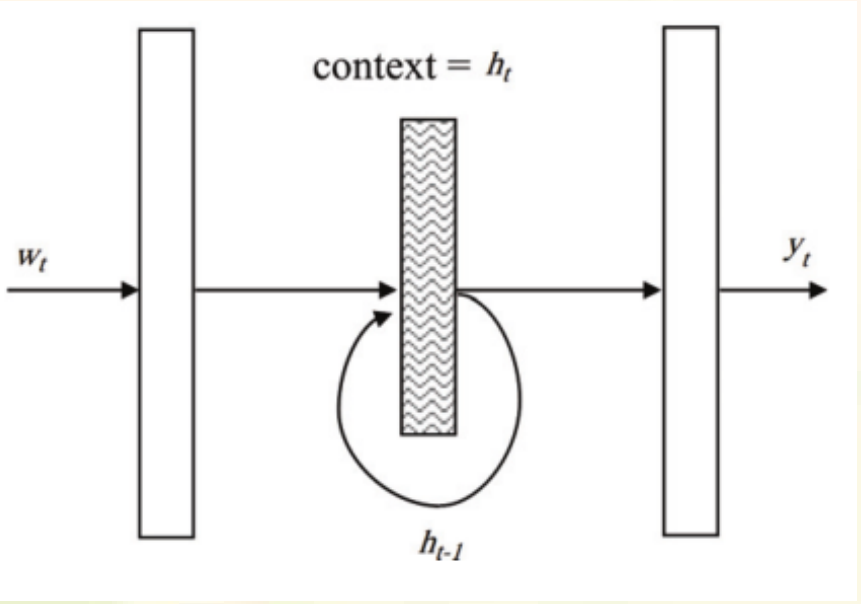
$$h_t = \phi(\mathbf{U}x_t). \quad (1)$$

$$y_t = g(\mathbf{V}h_t). \quad (2)$$

where ϕ is the sigmoid function and g is the softmax function:

$$\phi(z) = \frac{1}{1+e^{-z}}. \quad (3)$$

$$g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}. \quad (4)$$



$$y_i(j) = P(w_j | w_i, h_{i-1}).$$

Prior Knowledge Learning

log(D/Dw)

$$\mathrm{IDF}_{w_i} = log \frac{|N_D|}{|d \in D : w_i \in d|}.$$

$$\beta_i = \epsilon \times y_i(j) \times \mathrm{IDF}_{w_i}.$$

$$\beta_j = \epsilon \times y_i(j) \times \mathrm{IDF}_{w_j}.$$

$$b = (w_i, w_j, r_{ij}), \text{ where } r_{ij} = \langle \mathrm{IDF}_{w_i}, \mathrm{IDF}_{w_j}, y_i(j) \rangle.$$

RIBS-TM

1. Learn prior knowledge β from corpus D.
2. Draw $\theta \sim$ Dirichlet (α).
3. For each topic $k \in [1, K]$
   (a) draw $\phi_{k,w_i} \sim$ Dirichlet (β_i).
   (b) draw $\phi_{k,w_j} \sim$ Dirichlet (β_j).
4. For each biterm $b \in \mathbf{B}$, where $b = (w_i, w_j)$
   (a) draw $z \sim$ Multinomial (θ).
   (b) draw $w_i \sim$ Multinomial $(\phi_{z_{w_i}})$.
   draw $w_j \sim$ Multinomial $(\phi_{z_{w_j}})$.

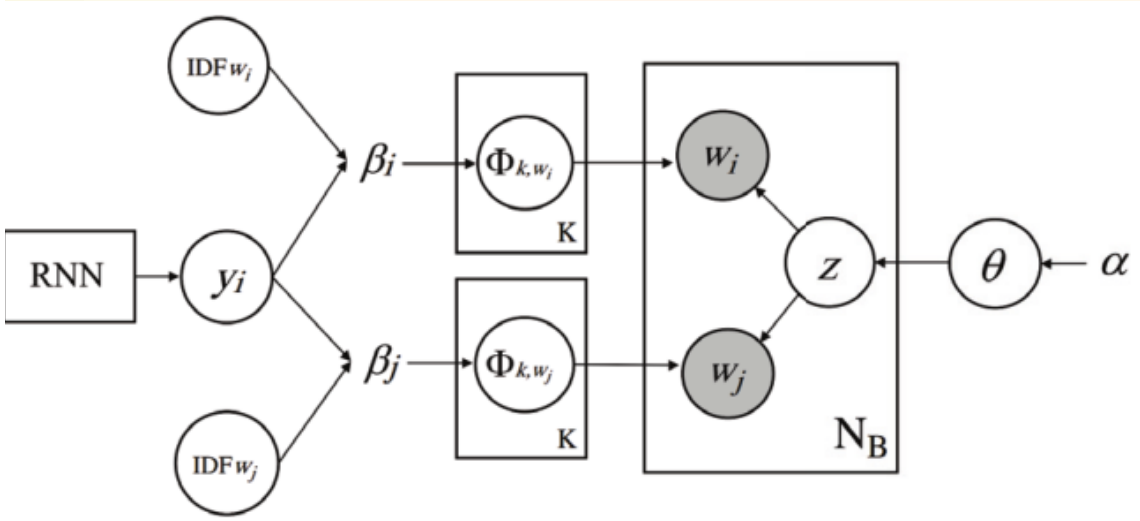


Figure 2: Graphical representation of RIBS-TM

Gibbs sampling

According to the chain rule on the joint probability of the corpus, we acquire the following conditional probability equation

$$p(z|z_{-b},\mathbf{B}) \propto \frac{(n_{-b,z}+\alpha)}{N_B+K\alpha}\frac{(n_{-b,w_i|z}+\beta_i)(n_{-b,w_j|z}+\beta_j)}{(\sum_w(n_{-b,w|z}+\beta))^2}.$$

$n_{-b,z}$ 为主题 z 中除去b后词对的数量

$n_{-b,w_i|z}$ 为主题 z 中除去b后单词wi的数量

$$\theta_k = \frac{(n_{z_k}+\alpha)}{N_B+K\alpha}.$$

$$\phi_{k,w_i} = \frac{n_{w_i|z_k}+\beta_i}{\sum_w(n_{w|z_k}+\beta)}.$$

$$\phi_{k,w_j} = \frac{n_{w_j|z_k}+\beta_j}{\sum_w(n_{w|z_k}+\beta)}.$$

Gibbs sampling

Algorithm 1 Gibbs sampling algorithm for RIBS-TM

Input: topic number K, α, β, biterm set $\mathbf{B}$.
Output: θ and ϕ.

```
Initialize topic assignments for each biterm randomly.
for iter ← 1 to N_iter do
    for each biterm b = (w_i, w_j, r_ij) ∈ B do
        Draw topic z_k from P(z|z_{-b}, B).
        Update n_{z_k}, n_{w_i|z_k}, n_{w_j|z_k}.
    end for
end for
Compute θ by Eq. (10) and φ by Eq. (11)(12).
```

RIBS-TM

1. Learn prior knowledge β from corpus D.
2. Draw $\theta \sim$ Dirichlet (α).
3. For each topic $k \in [1, K]$
   (a) draw $\phi_{k,w_i} \sim$ Dirichlet (β_i).
   (b) draw $\phi_{k,w_j} \sim$ Dirichlet (β_j).
4. For each biterm $b \in \mathbf{B}$, where $b = (w_i, w_j)$
   (a) draw $z \sim$ Multinomial (θ).
   (b) draw $w_i \sim$ Multinomial $(\phi_{z_{w_i}})$.
   draw $w_j \sim$ Multinomial $(\phi_{z_{w_j}})$.

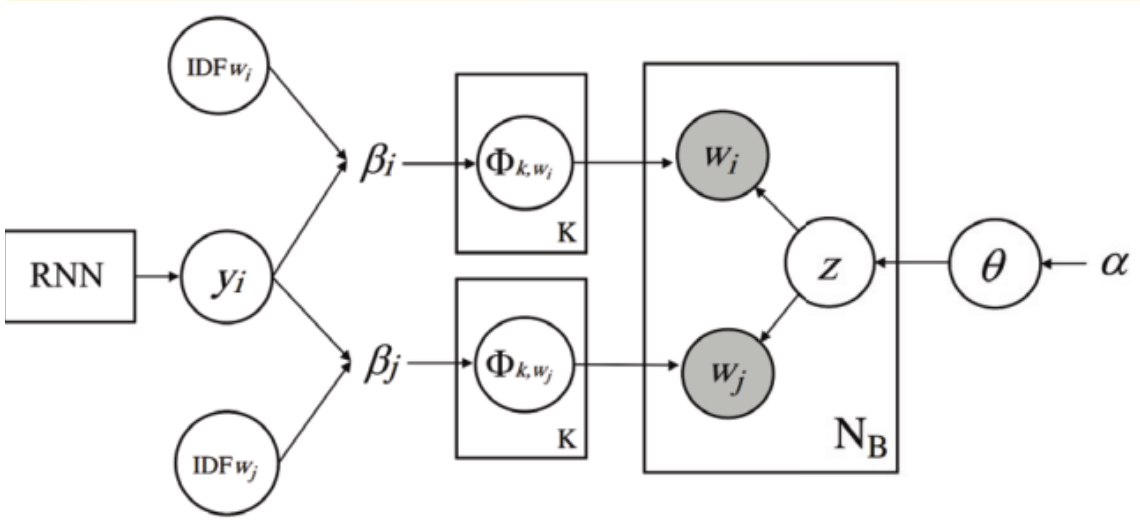


Figure 2: Graphical representation of RIBS-TM

Topics Inference

$$P(z_k|d) = \sum_{b \in \mathbf{B}} P(z_k, b|d) = \sum_{b \in \mathbf{B}} P(z_k|b, d)P(b|d).$$

由于 $P(z_k|b, d) = P(z_k|b)$

$$P(z_k|d) = \sum_{b \in \mathbf{B}} P(z_k|b)P(b|d).$$

$$P(z_k|b) = \frac{P(z_k)P(w_i|z_k)P(w_j|z_k)}{\sum_{k' \in K} P(z_{k'})P(w_i|z_{k'})P(w_j|z_{k'})}.$$

$P(z_k) = \theta_k, P(w_i|z_k) = \phi_{k,w_i}$

$$P(b|d) = \frac{n_d(b)}{\sum_{b \in \mathbf{B}} n_d(b)}.$$

Outputs of RIBS-TM are the $N_D \times K$ matrix for topic distribution over document and the $K \times W$ matrix for word distribution over topic, calculated as follows:

$$P(z|D) = [P(z|d_1), P(z|d_2), ..., P(z|d_{N_D})]$$

$$\phi = [\phi_{z_1}, \phi_{z_2}, ..., \phi_{z_K}] \qquad \phi_k = [\phi_{k,w_1}, \phi_{k,w_2}, \cdots, \phi_{k,w_W}]$$

Experiments

coherence

$$C = \frac{1}{K}\sum_{z=1}^{K}\sum_{m=2}^{M}\sum_{l=1}^{m-1} log\frac{n_D(w_m^z, w_l^z) + \epsilon'}{n_D(w_l^z)}.$$

Table 1: Coherence of Questions

	M=5	M=10	M=20
LDA	-38.4 ± 0.8	-216.7 ± 1.6	-1097.0 ± 4.2
BTM	-19.4 ± 0.6	-116.0 ± 1.8	-644.3 ± 1.6
d-BTM	-20.5 ± 0.2	-119.9 ± 1.7	-663.4 ± 0.7
RIBS-TM	**-17.6 ± 0.5**	**-105.3 ± 0.9**	**-601.7 ± 2.0**

Table 2: Coherence of News Titles

	M=5	M=10	M=20
LDA	-30.9 ± 0.6	-186.8 ± 2.4	-995.7 ± 5.2
BTM	-18.0 ± 0.4	-115.0 ± 1.6	-639.8 ± 3.9
d-BTM	-18.2 ± 0.2	-115.7 ± 2.3	-650.8 ± 1.8
RIBS-TM	**-16.4 ± 0.2**	**-106.3 ± 0.9**	**-602.4 ± 3.9**

谢谢