前沿技术讲习班
Advanced Technology Tutorial

深度学习与社会计算

李晨亮
软件工程国家重点实验室，计算机学院
武汉大学

讲习安排

- ✧ 社会计算简介（李晨亮）
- ✧ 基于社交媒体的社会计算（李晨亮）
  - ■ 用户画像，谣言预测
  - ■ 用户产生文本的特征表示
  - ■ 用户产生文本的语义理解
  - ■ 面向用户理解的个性化在线服务算法
    - ✓ 在线广告、标签预测、搜索多样化、人机对话
- ✧ 基于深度学习的网络表征（赵鑫）
  - ■ 网络表征模型
  - ■ 网络表征的应用
    - ✓ 可视化、文本分类、推荐系统
- ✧ 基于深度学习的推荐系统（赵鑫）
  - ■ 广告点击、音乐推荐、POI推荐

社会计算简介

- ✧ 社交媒体+移动通信技术
  - ■ 爆炸式的数据增长
  - ■ >75%的数据来自个人用户(麦肯锡全球研究院)
- ✧ 社会计算
  - ■ **大数据环境下个人、组织和社会的交往方式和相互关系，及其对人类社会不同文化群体和社会结构所造成的影响。**
  - ■ 社会信息 ➔➔ 社会智能
- ✧ 主要应用领域
  - ■ 社会与公共安全
  - ■ 商业智能
  - ■ 企业舆情分析
    - ✓ 在线广告、危机公关、服务多样化

社会计算简介

✧ 社会计算的关键技术方法

- ■ 海量用户产生信息的智能化感知
  - ✓ 社会信息的精准、实时和智能化获取
- ■ 多源异构社会信息的深刻理解
  - ✓ 语义提取与检索、多模态信息融合
- ■ 基于海量社会信息的人工社会构建
  - ✓ 社会网络关系、个体行为、智能化交互

✧ 面临的挑战

- ■ 社会信息的多样性与复杂性
  - ✓ 高纬度、噪声多、质量差和可信度低
- ■ 社会信息实时感知与语义鸿沟
  - ✓ 非结构化、个性化与多模态
- ■ 大规模社会群体行为的动态多样性
  - ✓ 社会舆情分析、个体观念

用户画像

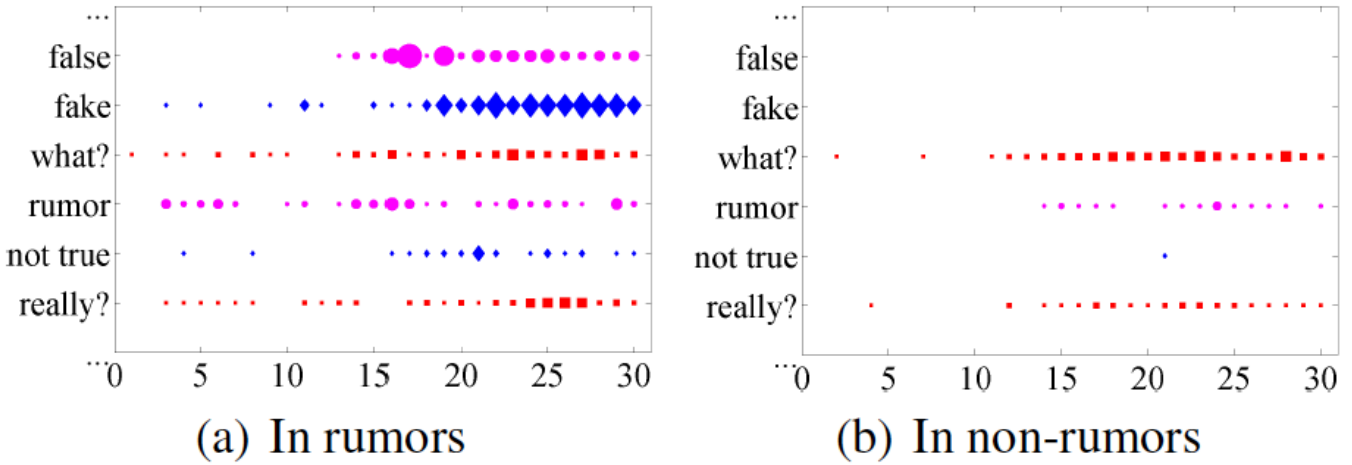
- 基于社交媒体用户产生文本
  - 用户所涉及的文本数据
  - 用户交互行为： 点赞、转发、评论
  - 主题模型技术（微博推荐、专家找寻、新闻摘要）
  - 隐因子模型（商品推荐、文章推荐、流行度预测）
  - 检索模型（微博检索、事件检测）
- 基于社交媒体用户交互关系
  - 用户之间的交互关系：朋友、关注、转发与点赞等
  - 常常作为辅助知识与文本信息统一建模
    - 作为隐因子模型的约束项、主题模型平滑
  - 用户多角色建模
  - 用户关系预测、用户关系分析
  - 专家找寻

用户文本理解及个性化服务

- ✧ 用户文本理解
  - ■ 用户发表的评论，微博
    - ✓ 观点抽取、事件检测，用户个人信息的暴露
  - ■ 用户交互内容
    - ✓ 聊天、转发，
    - ✓ 人机聊天，用户社区发现
  - ■ 元数据信息语义构建
    - ✓ Hashtag推荐，文本分类，用户分类
- ✧ 传统方法的挑战
  - ■ Bag-of-Word词袋模型
    - ✓ 丢失单词组合顺序、歧义性
  - ■ One-Hot特征表达
    - ✓ 相关信息丢失，维度灾难
  - ■ 文本共生信息缺失（短文本）
    - ✓ 上下文信息缺乏、主题模型效果较差

微博谣言预测

- ✧ 谣言的危害
  - ■ 引起社会民众的恐慌与群体暴动
    - ✓ “食盐防辐射”谣言导致抢盐风波
    - ✓ 韦拉克鲁斯学校枪战儿童绑架
  - ■ 人工检测谣言
    - ✓ 常识知识与专家咨询
    - ✓ Snopes.com众包服务
    - ✓ 范围较局限、响应时间较长
- ✧ 现有的传统方法
  - ■ 监督型机器学习算法
    - ✓ SVM、Decision Tree、Random Forest
  - ■ 人工特征设计
    - ✓ 微博数量统计、用户和传播特征抽取
    - ✓ 微博用户评论内容分析、关键字分析
    - ✓ 时间序列特征、时序统计特征



微博谣言预测

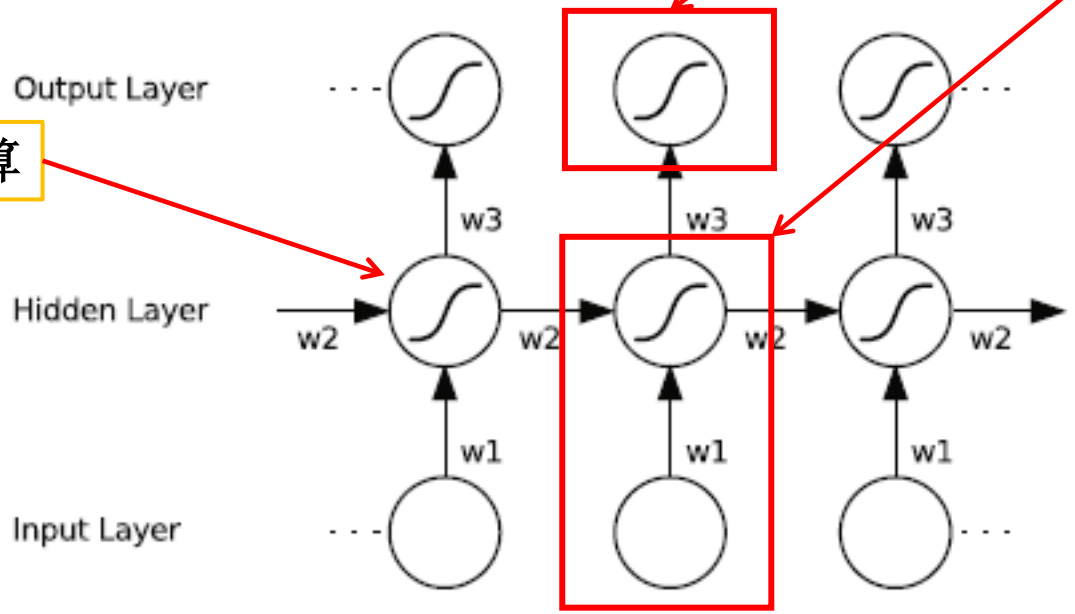
- 基于RNN模型的微博谣言预测 [Ma et al. IJCAI 2016]
  - RNN介绍
    - 对变长的序列数据进行建模和特征抽取
    - 基于当前输入与上一个时刻隐状态更新当前隐状态

当前隐状态作为当前时刻的特征表达

$$h_t = tanh(Ux_t + Wh_{t-1} + b)$$

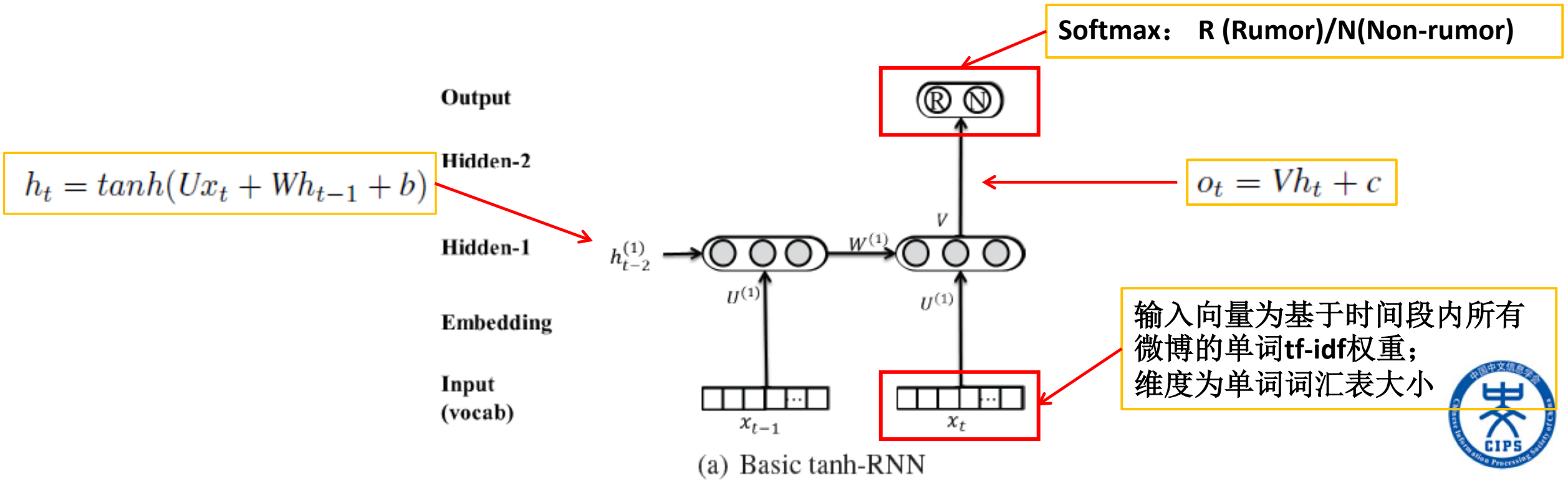
基于当前输入与上一个时刻隐状态计算当前隐状态向量

上一时刻的隐状态计算



微博谣言预测

- 基于RNN模型的微博谣言预测 [Ma et al. IJCAI 2016]
  - 给定一个事件 *E*，及其相关微博 *E*={(m_i, t_i)}: m_i代表某条具体微博、t_i为微博的发布时间
  - 将事件相关微博按照时间间隔划分成基于时间间隔的序列



微博谣言预测

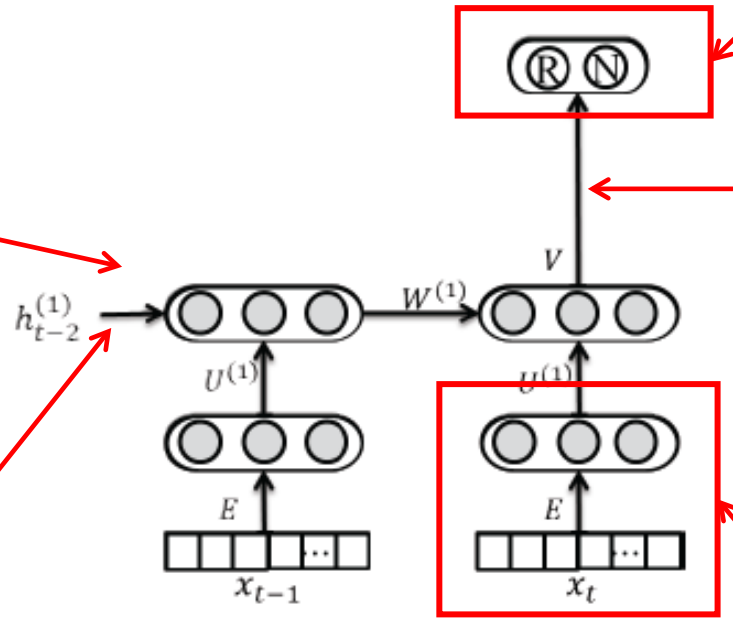
✧ 基于RNN模型的微博谣言预测 [Ma et al. IJCAI 2016]

LSTM

$$
\begin{aligned}
i_t &= \sigma(x_t W_i + h_{t-1} U_i + c_{t-1} V_i) \\
f_t &= \sigma(x_t W_f + h_{t-1} U_f + c_{t-1} V_f) \\
\tilde{c}_t &= tanh(x_t W_c + h_{t-1} U_c) \\
c_t &= f_t c_{t-1} + i_t \tilde{c}_t \\
o_t &= \sigma(x_t W_o + h_{t-1} U_o + c_t V_o) \\
h_t &= o_t \, tanh(c_t)
\end{aligned}
$$

GRU

$$
\begin{aligned}
z_t &= \sigma(x_t U_z + h_{t-1} W_z) \\
r_t &= \sigma(x_t U_r + h_{t-1} W_r) \\
\tilde{h}_t &= tanh(x_t U_h + (h_{t-1} \cdot r_t) W_h) \\
h_t &= (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t
\end{aligned}
$$



(b) 1-layer LSTM/GRU + embedding

Softmax： R (Rumor)/N(Non-rumor)

$$o_t = Vh_t + c$$

$$x_e = x_t E$$

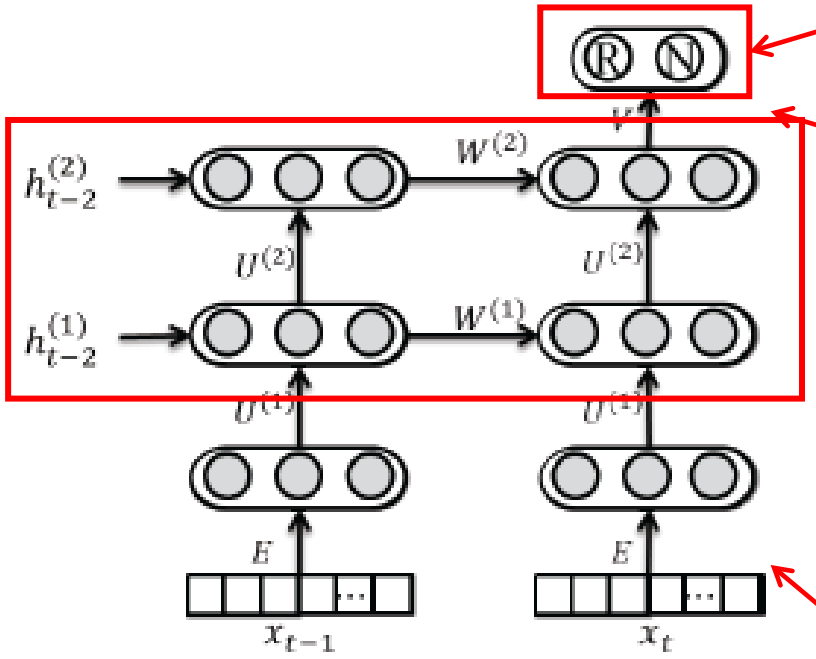
输入向量为基于时间段内所有微博的单词tf-idf权重；
通过词向量矩阵转化到词向量空间；
维度为100

微博谣言预测

✧ 基于RNN模型的微博谣言预测 [Ma et al. IJCAI 2016]

$$z_t^{(1)} = \sigma\left(x_e U_z^{(1)} + h_{t-1}^{(1)} W_z^{(1)}\right)$$
$$r_t^{(1)} = \sigma\left(x_e U_r^{(1)} + h_{t-1}^{(1)} W_r^{(1)}\right)$$
$$\tilde{h}_t^{(1)} = tanh\left(x_e U_h^{(1)} + (h_{t-1}^{(1)} \cdot r_t^{(1)}) W_h^{(1)}\right)$$
$$h_t^{(1)} = (1 - z_t^{(1)}) \cdot h_{t-1}^{(1)} + z_t^{(1)} \cdot \tilde{h}_t^{(1)}$$

$$z_t^{(2)} = \sigma\left(h_t^{(1)} U_z^{(2)} + h_{t-1}^{(2)} W_z^{(2)}\right)$$
$$r_t^{(2)} = \sigma\left(h_t^{(1)} U_r^{(2)} + h_{t-1}^{(2)} W_r^{(2)}\right)$$
$$\tilde{h}_t^{(2)} = tanh\left(h_t^{(1)} U_h^{(2)} + (h_{t-1}^{(2)} \cdot r_t^{(2)}) W_h^{(2)}\right)$$
$$h_t^{(2)} = (1 - z_t^{(2)}) \cdot h_{t-1}^{(2)} + z_t^{(2)} \cdot \tilde{h}_t^{(2)}$$



(c) 2-layer GRU + embedding

Softmax： R (Rumor)/N(Non-rumor)

$$o_t = Vh_t + c$$

$$x_e = x_t E$$

输入向量为基于时间段内所有微博的单词tf-idf权重；
通过词向量矩阵转化到词向量空间；维度为100

微博谣言预测

✧ 基于RNN模型的微博谣言预测 [Ma et al. IJCAI 2016]

- 模型训练（Squared Error Minimization）

$$min \quad \sum_{c} (g_c - p_c)^2 + \sum_{i} ||\theta_i||^2$$

参数L2 正则化

真实结果分布

预测结果分布

- 两个微博数据集：1）Twitter； 2）Sina Weibo

Table 1: Statistics of the datasets

Statistic	Twitter	Weibo
Users #	491,229	2,746,818
Posts #	1,101,985	3,805,656
Events #	992	4,664
Rumors #	498	2,313
Non-Rumors #	494	2,351
Avg. time length / event	1,582.6 Hours	2,460.7 Hours
Avg. # of posts / event	1,111	816
Max # of posts / event	62,827	59,318
Min # of posts / event	10	10

微博谣言预测

✧ 现有对比方法

- SVM-TS [Ma et al CIKM 2015]
  - ✓ 动态时间序列建模
  - ✓ 文本内容、用户和传播模式
- DT-Rank [Zhao et al WWW 2015]
  - ✓ 提示词: “not true”, “unconfirmed”, “debunk”
  - ✓ 基于提示词检索质疑和否定微博相关内容
  - ✓ 基于相关微博统计特征构建决策树分类器
- DTC/SVM-RBF [Castillo et al WWW 2011]
  - ✓ 相关微博的统计特征
- RFC [Kwon et al ICDM 2013]
  - ✓ 基于时序微博特征构建随机森林分类器

人工构建基于时间序列、特征词和文本信息的特征

微博谣言预测

✧ 实验性能对比

- Accuracy, Precision, Recall, F1

基于**RNN**的模型相比传统特征提取的方法有较大的性能优势

(a) Twitter dataset

Method	Class	Accuracy	Precision	Recall	F_1
DT-Rank	R	0.644	0.638	0.675	0.656
	N		0.652	0.613	0.632
SVM-RBF	R	0.722	0.856	0.526	0.651
	N		0.663	**0.914**	0.769
DTC	R	0.731	0.724	0.757	0.740
	N		0.739	0.704	0.721
RFC	R	0.772	0.717	0.908	0.801
	N		0.870	0.634	0.734
SVM-TS	R	0.808	0.735	**0.963**	0.834
	N		**0.947**	0.652	0.772
tanh-RNN	R	0.827	0.847	0.833	0.840
	N		0.804	0.820	0.812
LSTM-1	R	0.855	0.855	0.883	0.869
	N		0.854	0.820	0.837
GRU-1	R	0.864	**0.857**	0.900	0.878
	N		0.872	0.820	0.845
GRU-2	R	**0.881**	0.851	0.950	**0.898**
	N		0.930	0.800	**0.860**

(b) Weibo dataset

Method	Class	Accuracy	Precision	Recall	F_1
DT-Rank	R	0.732	0.738	0.715	0.726
	N		0.726	0.749	0.737
SVM-RBF	R	0.818	0.822	0.812	0.817
	N		0.815	0.824	0.819
DTC	R	0.831	0.847	0.815	0.831
	N		0.815	0.847	0.830
RFC	R	0.849	0.786	0.959	0.864
	N		0.947	0.739	0.830
SVM-TS	R	0.857	0.839	0.885	0.861
	N		0.878	0.830	0.857
tanh-RNN	R	0.873	0.816	0.964	0.884
	N		**0.956**	0.782	0.861
LSTM-1	R	0.896	0.846	**0.968**	0.913
	N		0.953	0.858	0.903
GRU-1	R	0.908	0.871	0.958	0.913
	N		0.953	0.858	0.903
GRU-2	R	**0.910**	**0.876**	0.956	**0.914**
	N		0.952	**0.864**	**0.906**

微博谣言预测

✧ 实验性能对比

- Accuracy, Precision, Recall, F1

基于**GRU**的两层模型具有更加的**F1**性能指标

(a) Twitter dataset

Method	Class	Accuracy	Precision	Recall	F_1
DT-Rank	R	0.644	0.638	0.675	0.656
	N		0.652	0.613	0.632
SVM-RBF	R	0.722	0.856	0.526	0.651
	N		0.663	**0.914**	0.769
DTC	R	0.731	0.724	0.757	0.740
	N		0.739	0.704	0.721
RFC	R	0.772	0.717	0.908	0.801
	N		0.870	0.634	0.734
SVM-TS	R	0.808	0.735	**0.963**	0.834
	N		**0.947**	0.652	0.772
tanh-RNN	R	0.827	0.847	0.833	0.840
	N		0.804	0.820	0.812
LSTM-1	R	0.855	0.855	0.883	0.869
	N		0.854	0.820	0.837
GRU-1	R	0.864	**0.857**	0.900	0.878
	N		0.872	0.820	0.845
GRU-2	R	**0.881**	0.851	0.950	**0.898**
	N		0.930	0.800	**0.860**

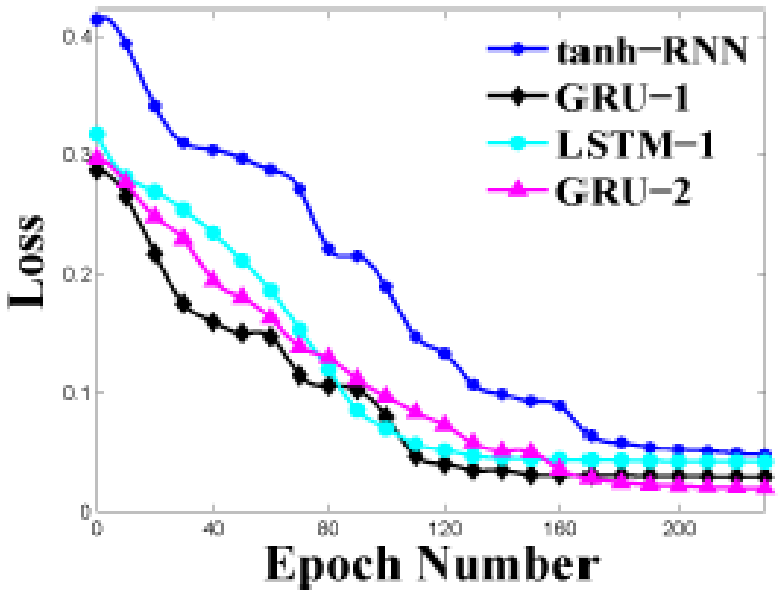
(b) Weibo dataset

Method	Class	Accuracy	Precision	Recall	F_1
DT-Rank	R	0.732	0.738	0.715	0.726
	N		0.726	0.749	0.737
SVM-RBF	R	0.818	0.822	0.812	0.817
	N		0.815	0.824	0.819
DTC	R	0.831	0.847	0.815	0.831
	N		0.815	0.847	0.830
RFC	R	0.849	0.786	0.959	0.864
	N		0.947	0.739	0.830
SVM-TS	R	0.857	0.839	0.885	0.861
	N		0.878	0.830	0.857
tanh-RNN	R	0.873	0.816	0.964	0.884
	N		**0.956**	0.782	0.861
LSTM-1	R	0.896	0.846	**0.968**	0.913
	N		0.953	0.858	0.903
GRU-1	R	0.908	0.871	0.958	0.913
	N		0.953	0.858	0.903
GRU-2	R	**0.910**	**0.876**	0.956	**0.914**
	N		0.952	**0.864**	**0.906**

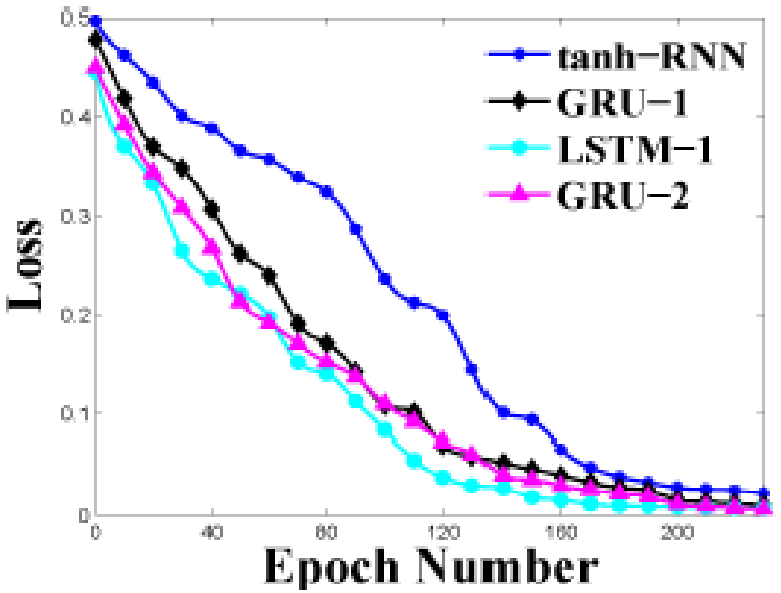
微博谣言预测

✧ 实验性能对比

- Accuracy, Precision, Recall, F1



(a) Twitter dataset



(b) Weibo dataset

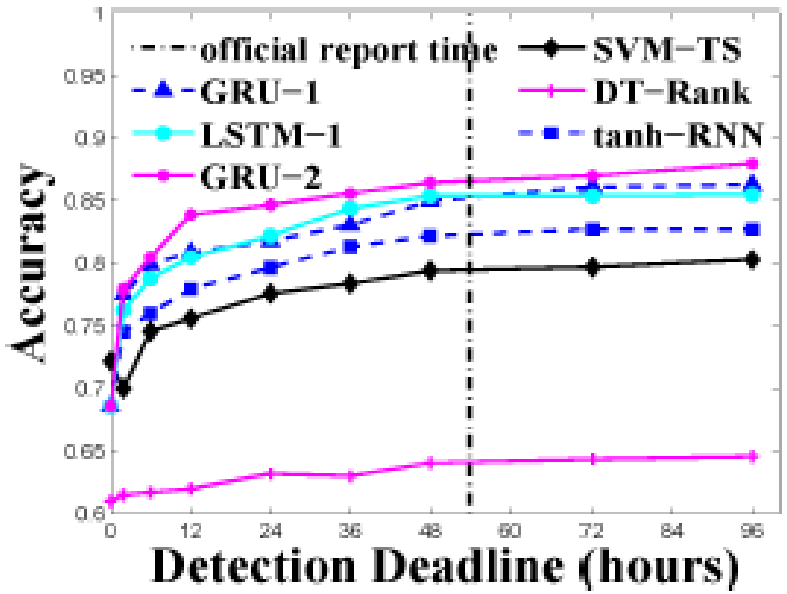
基于GRU/LSTM的模型收敛地更快

微博谣言预测

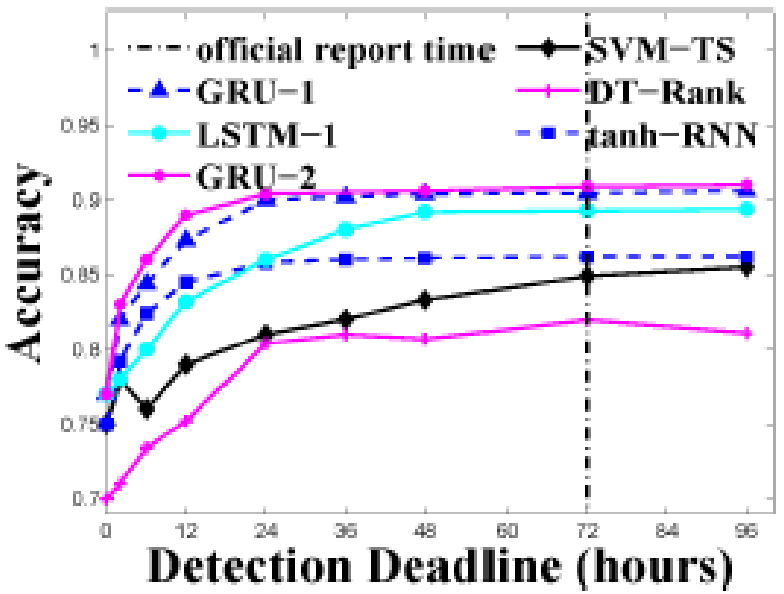
✧ 实验性能对比

- Accuracy, Precision, Recall, F1

面对少量数据，两层GRU模型具有最佳的预测性能



(a) Twitter dataset

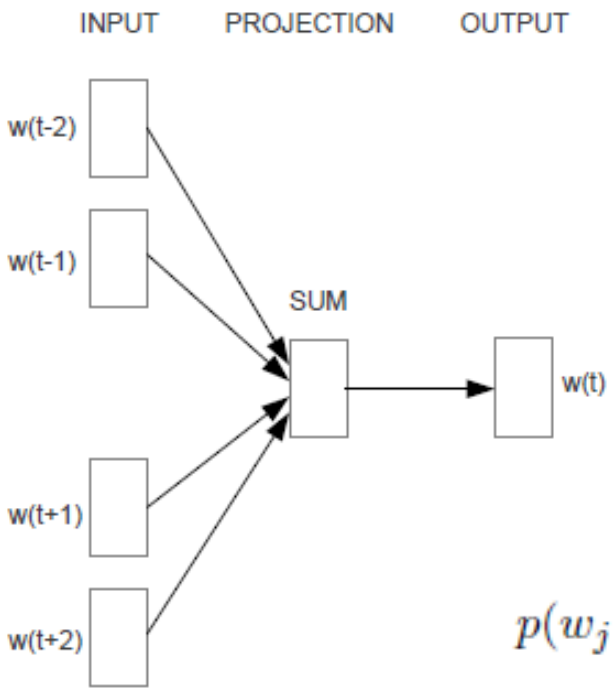


(b) Weibo dataset

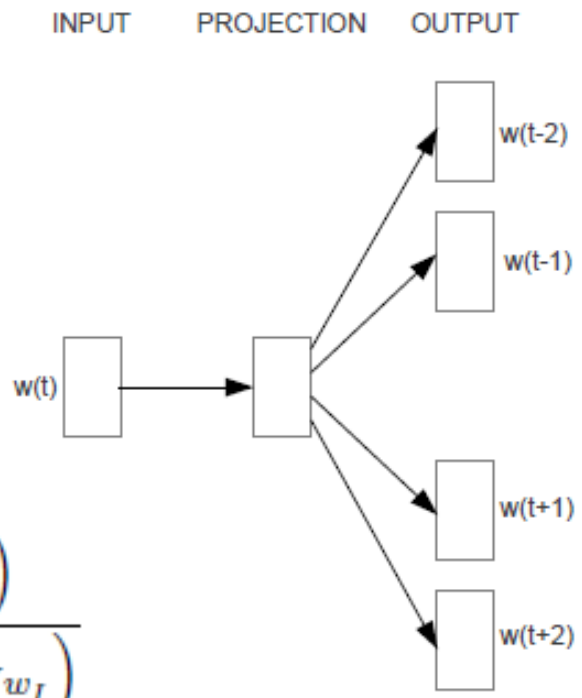
仅仅提供deadline之前的数据

社交媒体用户画像 – 文本Embedding技术

- CBOW/SkipGram [Mikolov et al arXiv 2013]



$$p(w_j|w_I) = \frac{\exp\left({\mathbf{v}'_{w_O}}^T \mathbf{v}_{w_I}\right)}{\sum_{j'=1}^{V} \exp\left({\mathbf{v}'_{w'_j}}^T \mathbf{v}_{w_I}\right)}$$



CBOW

$\log P(w_t|w_{t-k},\ldots,w_{t+k})$
基于上下文窗口单词预测中间（当前）单词

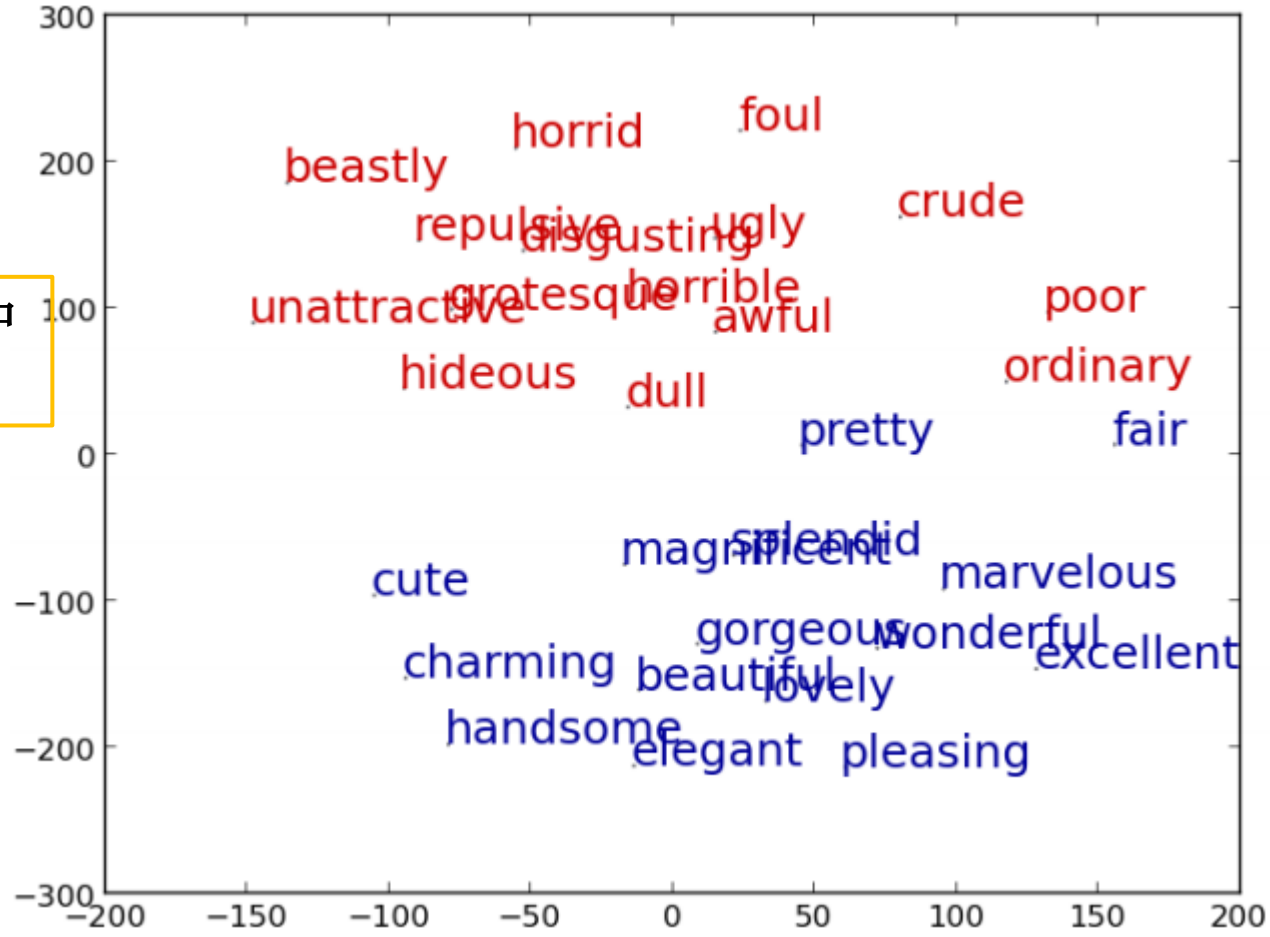
Skip-gram

$\log P(w_{t+j}|w_t)$
基于中间（当前）单词预测上下文窗口单词

社交媒体用户画像 – 文本Embedding技术

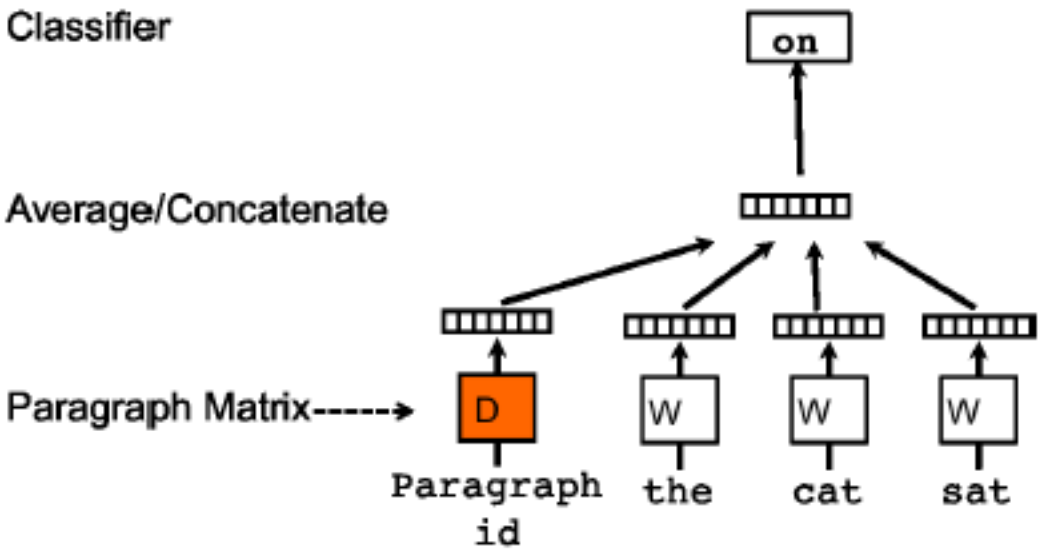
- CBOW/SkipGram [Mikolov et al arXiv 2013]

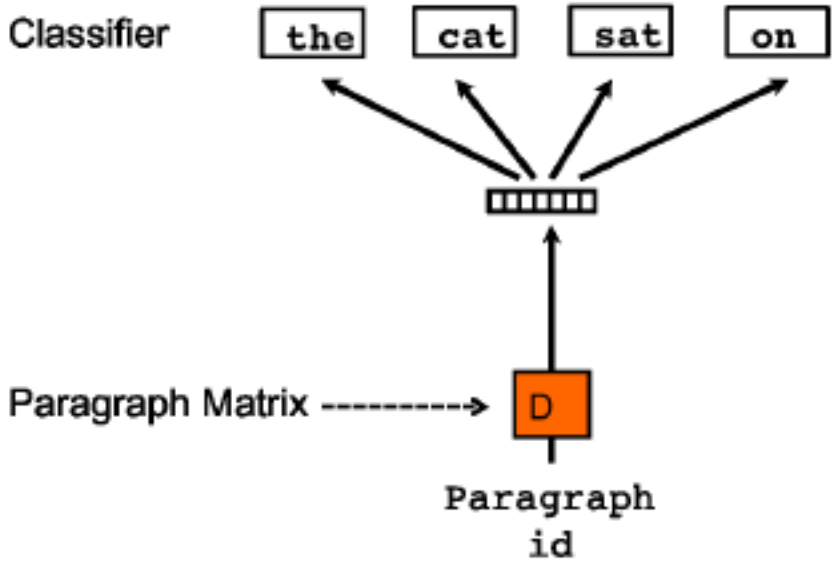
语义/语法作用相似的单词在隐空间中彼此接近



社交媒体用户画像 – 文本Embedding技术

✧ Paragraph Vector [Le and Mikolov arXiv 2014]



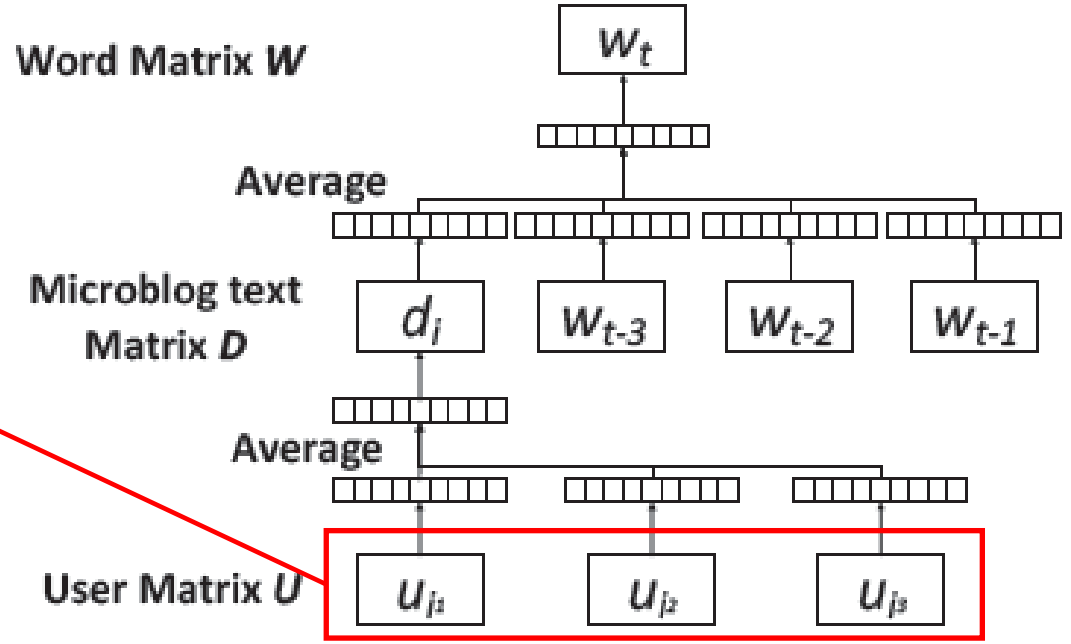


$\log P(w_t | d, w_{t-k}, \ldots, w_{t+k})$
基于上下文窗口单词和文本向量预测中间（当前）单词

$\log P(w_{t+j} | d)$
基于文本向量预测上下文窗口单词

社交媒体用户画像 – 文本Embedding技术

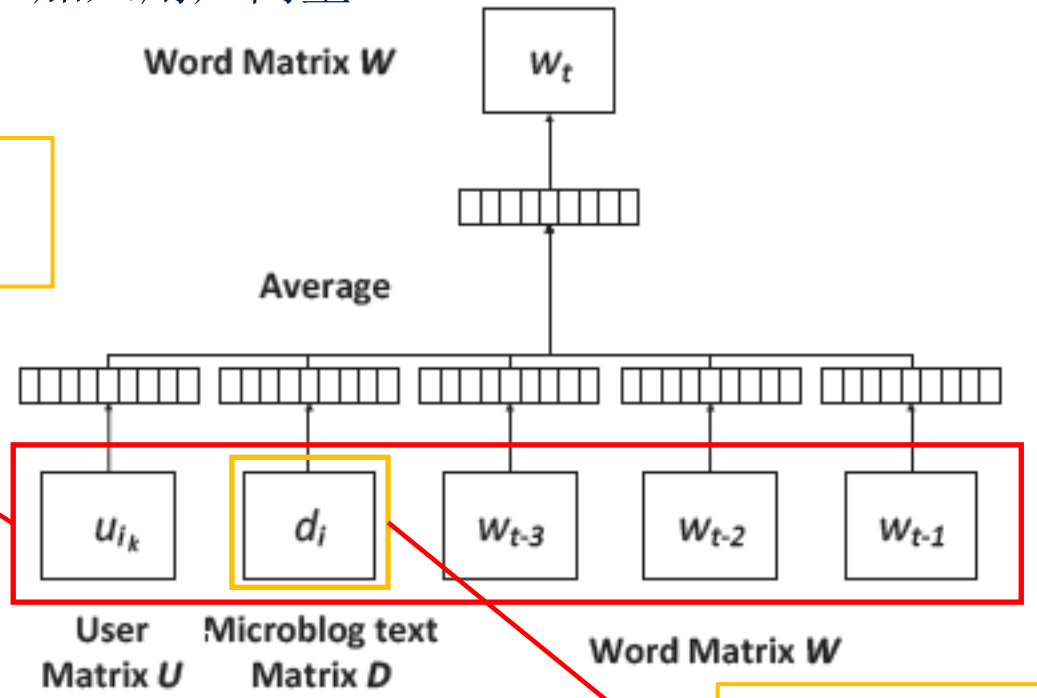
- 学术微博推荐中的用户Embedding技术 [Yu et al ACL 2016]
- 模型1 （User2Vec#1）：基于CBOW和Paragraph Vector
- 一层CBOW加上一层Paragraph Vector



发布、转发和评论该条微博的所有用户；
用户向量预测文本向量；
文本向量与上下文单词预测当前单词。

社交媒体用户画像 – 文本Embedding技术

- 学术微博推荐中的用户Embedding技术 [Yu et al ACL 2016]
- 模型2 （User2Vec#2）：基于CBOW和Paragraph Vector
- 对Paragraph Vector进行扩展，加入用户向量



用户向量、文本向量与上下文单词预测当前单词。

一条微博可以被多个用户关联，文本向量全局唯一。

社交媒体用户画像 – 文本Embedding技术

- 学术微博推荐中的用户Embedding技术 [Yu et al ACL 2016]
  - 用户向量与微博向量的余弦距离用来确定推荐微博排序
- 对比方法
  - BOW: 用词袋模型表达每个微博，用户向量为微博词袋模型的均值
  - SVM on BOW: 基于BOW，对每个用户构建SVM分类器
  - Average Embedding: 基于Paragraph Vector学习微博向量，用户向量为微博向量的均值

Average Embedding好于BOW的方法

	k=10			*k*=20			*k*=50			*k*=100		
	Precision	Recall	MRR	Precision	Recall	MRR	Precision	Recall	MRR	Precision	Recall	MRR
Bag-of-Words	0.5036	0.0504	0.0153	0.4917	0.0983	0.0185	0.4461	0.2231	0.0223	0.3204	0.3204	0.0246
SVM on BoW	0.5774	0.0577	0.0172	0.5662	0.1132	0.0212	0.5122	0.2561	0.0256	0.3675	0.3675	0.0282
Average Embedding	0.5963	0.0596	0.0183	0.5824	0.1165	0.0219	0.5266	0.2633	0.0264	0.3793	0.3793	0.0291
User2Vec#1	0.6246	0.0625	0.0189	0.6055	0.1211	0.0228	0.5511	0.2756	0.0275	0.3953	0.3953	0.0304
User2Vec#2	**0.6652**	**0.0665**	**0.0201**	**0.6498**	**0.1300**	**0.0244**	**0.5883**	**0.2942**	**0.0295**	**0.4231**	**0.4231**	**0.0325**

社交媒体用户画像 – 文本Embedding技术

- 学术微博推荐中的用户Embedding技术 [Yu et al ACL 2016]
  - 用户向量与微博向量的余弦距离用来确定推荐排序
- 对比方法
  - BOW: 用词袋模型表达每个微博，用户向量为微博词袋模型的均值
  - SVM on BOW: 基于BOW，对每个用户构建SVM分类器
  - Average Embedding: 基于Paragraph Vector学习微博向量，用户向量为微博向量的均值

对用户向量的单独估计的性能好于**Average Embedding**

	k=10			*k*=20			*k*=50			*k*=100		
	Precision	Recall	MRR	Precision	Recall	MRR	Precision	Recall	MRR	Precision	Recall	MRR
Bag-of-Words	0.5036	0.0504	0.0153	0.4917	0.0983	0.0185	0.4461	0.2231	0.0223	0.3204	0.3204	0.0246
SVM on BoW	0.5774	0.0577	0.0172	0.5662	0.1132	0.0212	0.5122	0.2561	0.0256	0.3675	0.3675	0.0282
Average Embedding	0.5963	0.0596	0.0183	0.5824	0.1165	0.0219	0.5266	0.2633	0.0264	0.3793	0.3793	0.0291
User2Vec#1	0.6246	0.0625	0.0189	0.6055	0.1211	0.0228	0.5511	0.2756	0.0275	0.3953	0.3953	0.0304
User2Vec#2	**0.6652**	**0.0665**	**0.0201**	**0.6498**	**0.1300**	**0.0244**	**0.5883**	**0.2942**	**0.0295**	**0.4231**	**0.4231**	**0.0325**

社交媒体用户画像 – 文本Embedding技术

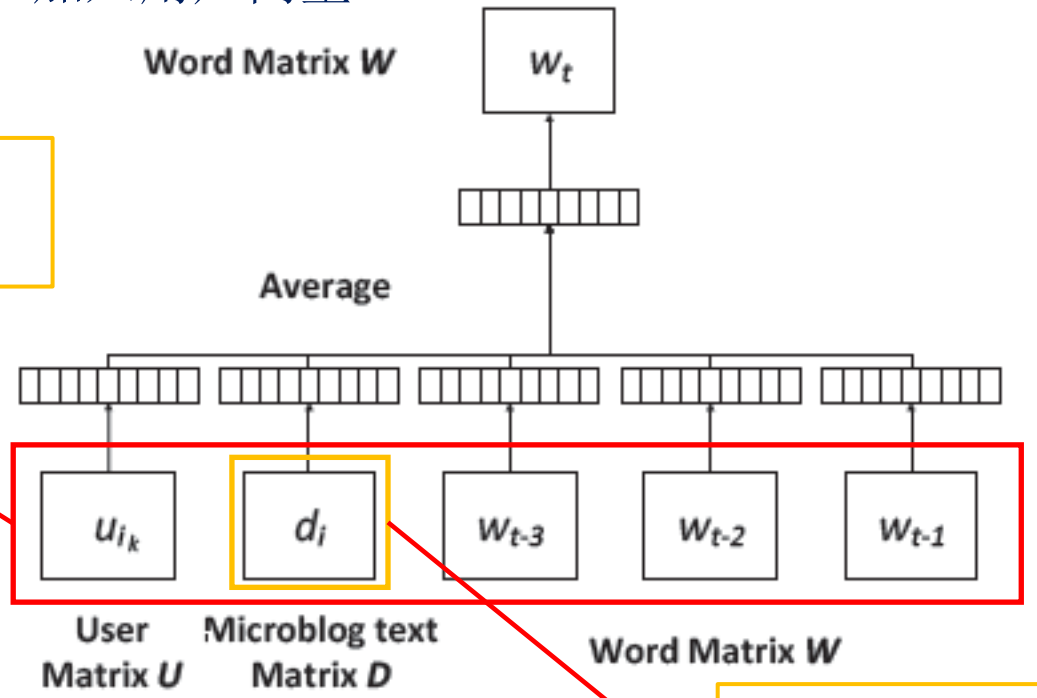
- 学术微博推荐中的用户Embedding技术 [Yu et al ACL 2016]
  - 用户向量与微博向量的余弦距离用来确定推荐排序
- 对比方法
  - BOW: 用词袋模型表达每个微博，用户向量为微博词袋模型的均值
  - SVM on BOW: 基于BOW，对每个用户构建SVM分类器
  - Average Embedding: 基于Paragraph Vector学习微博向量，用户向量为微博向量的均值

模型2更简单，性能更佳

	k=10			*k*=20			*k*=50			*k*=100		
	Precision	Recall	MRR	Precision	Recall	MRR	Precision	Recall	MRR	Precision	Recall	MRR
Bag-of-Words	0.5036	0.0504	0.0153	0.4917	0.0983	0.0185	0.4461	0.2231	0.0223	0.3204	0.3204	0.0246
SVM on BoW	0.5774	0.0577	0.0172	0.5662	0.1132	0.0212	0.5122	0.2561	0.0256	0.3675	0.3675	0.0282
Average Embedding	0.5963	0.0596	0.0183	0.5824	0.1165	0.0219	0.5266	0.2633	0.0264	0.3793	0.3793	0.0291
User2Vec#1	0.6246	0.0625	0.0189	0.6055	0.1211	0.0228	0.5511	0.2756	0.0275	0.3953	0.3953	0.0304
User2Vec#2	**0.6652**	**0.0665**	**0.0201**	**0.6498**	**0.1300**	**0.0244**	**0.5883**	**0.2942**	**0.0295**	**0.4231**	**0.4231**	**0.0325**

社交媒体用户画像 – 文本Embedding技术

- 学术微博推荐中的用户Embedding技术 [Yu et al ACL 2016]
- 模型2 （User2Vec#2）：基于CBOW和Paragraph Vector
- 对Paragraph Vector进行扩展，加入用户向量



用户向量、文本向量与上下文单词预测当前单词。

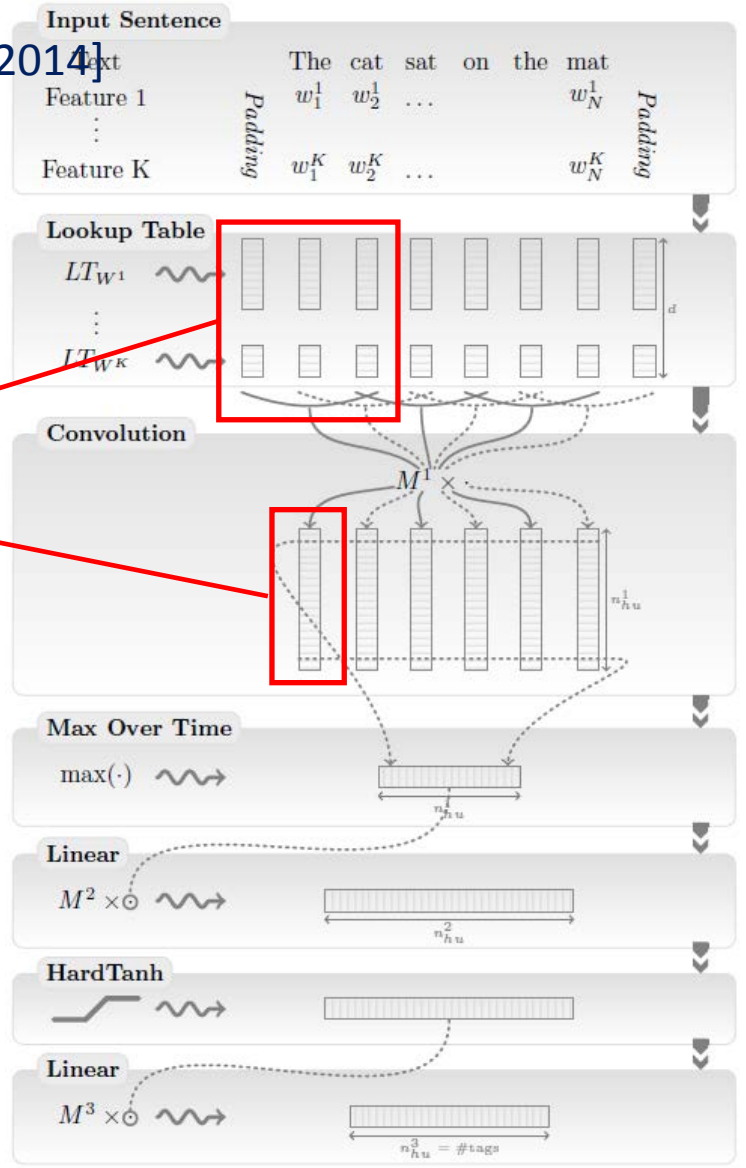
用户向量、文本向量与单词向量同时学习具有更优的表达性能。

一条微博可以被多个用户关联，文本向量全局唯一。

用户文本理解

- 借助Hashtag的微博文本语义Embedding [Weston et al EMNLP2014]
  - 运用CNN模型将单词、微博和Hashtag用同一向量空间表示
- CNN介绍
  - 对序列进行特征提取，将变长序列转换为固定长度特征向量
  - 窗口长度K滑动

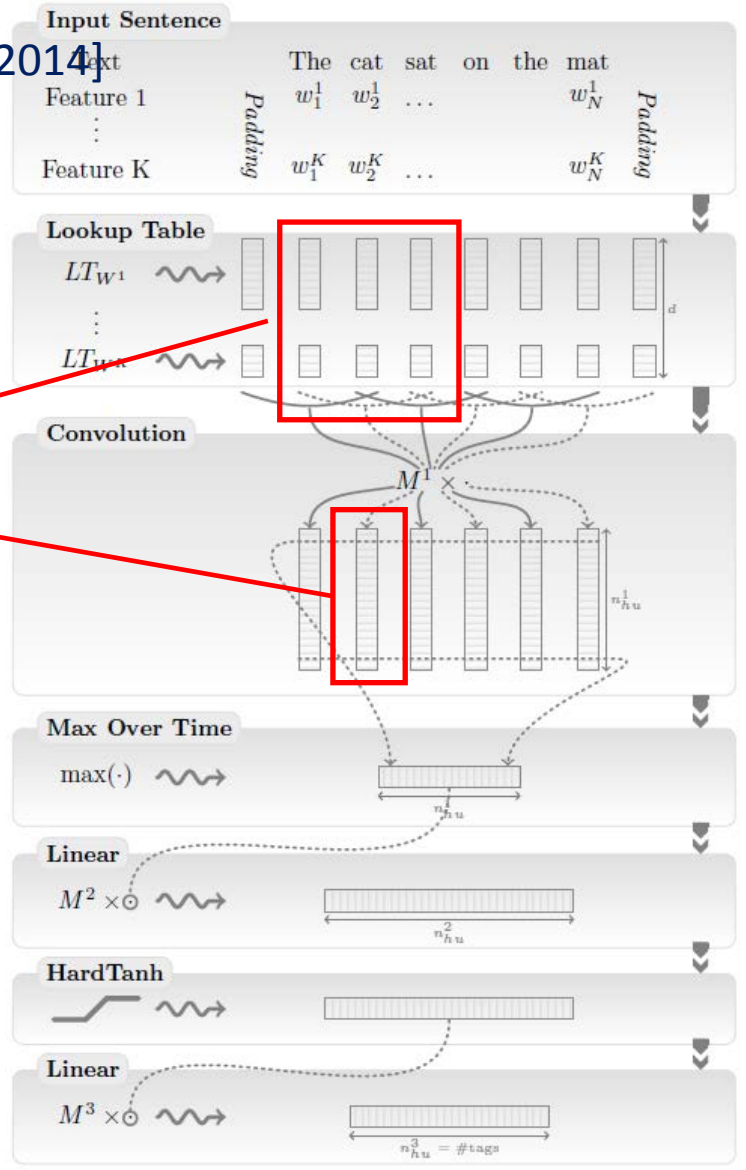
窗口K滑动抽取局部特征



用户文本理解

- 借助Hashtag的微博文本语义Embedding [Weston et al EMNLP2014]
  - 运用CNN模型将单词、微博和Hashtag用同一向量空间表示
- CNN介绍
  - 对序列进行特征提取，将变长序列转换为固定长度特征向量
  - 窗口长度K滑动

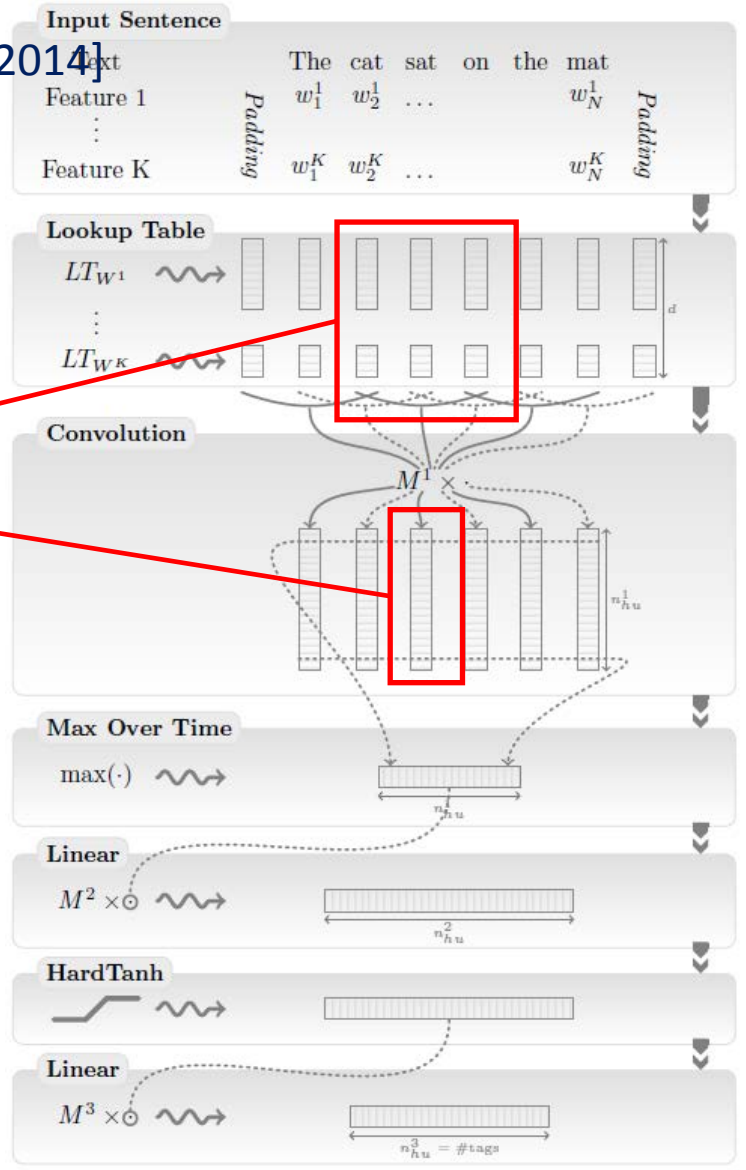
窗口K滑动抽取局部特征



用户文本理解

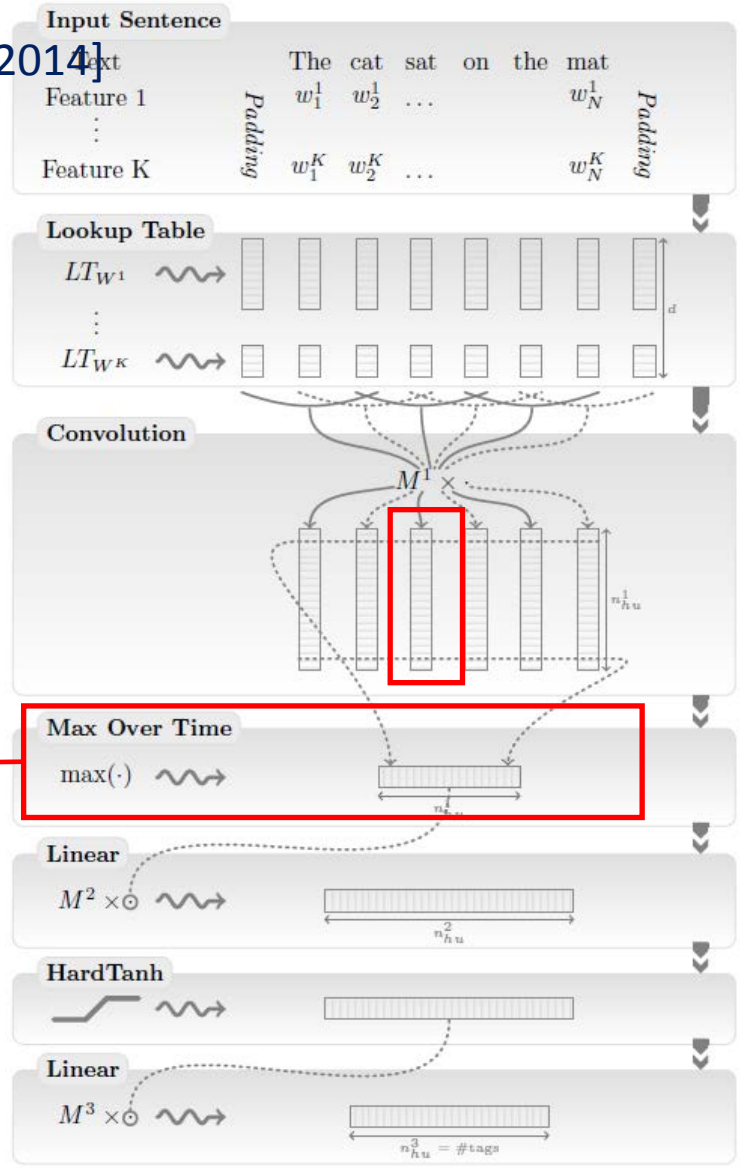
- 借助Hashtag的微博文本语义Embedding [Weston et al EMNLP2014]
  - 运用CNN模型将单词、微博和Hashtag用同一向量空间表示
- CNN介绍
  - 对序列进行特征提取，将变长序列转换为固定长度特征向量
  - 窗口长度K滑动

窗口K滑动抽取局部特征



用户文本理解

- 借助Hashtag的微博文本语义Embedding [Weston et al EMNLP2014]
  - 运用CNN模型将单词、微博和Hashtag用同一向量空间表示
- CNN介绍
  - 对序列进行特征提取，将变长序列转换为固定长度特征向量
  - 窗口长度K滑动
  - 池化过程通过局部特征构建全局特征



池化过程：基于每个维度的局部特征抽取全局特征。
常用方法：最大值、最小值、均值

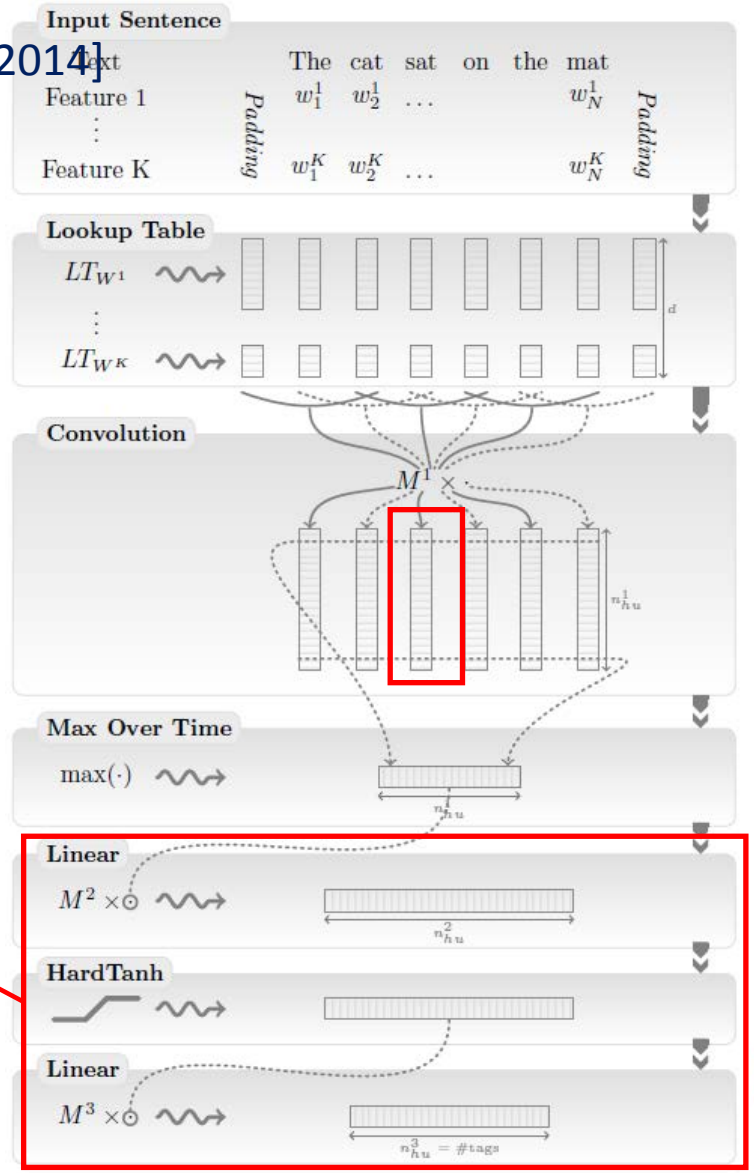
0.1	0.3	0.4	0.5
0.2	0.1	0.7	0.9

Max	Min	Mean
0.5	0.1	0.325
0.9	0.1	0.475

用户文本理解

- 借助Hashtag的微博文本语义Embedding [Weston et al EMNLP2014]
  - 运用CNN模型将单词、微博和Hashtag用同一向量空间表示
- CNN介绍
  - 对序列进行特征提取，将变长序列转换为固定长度特征向量
  - 窗口长度K滑动
  - 池化过程通过局部特征构建全局特征

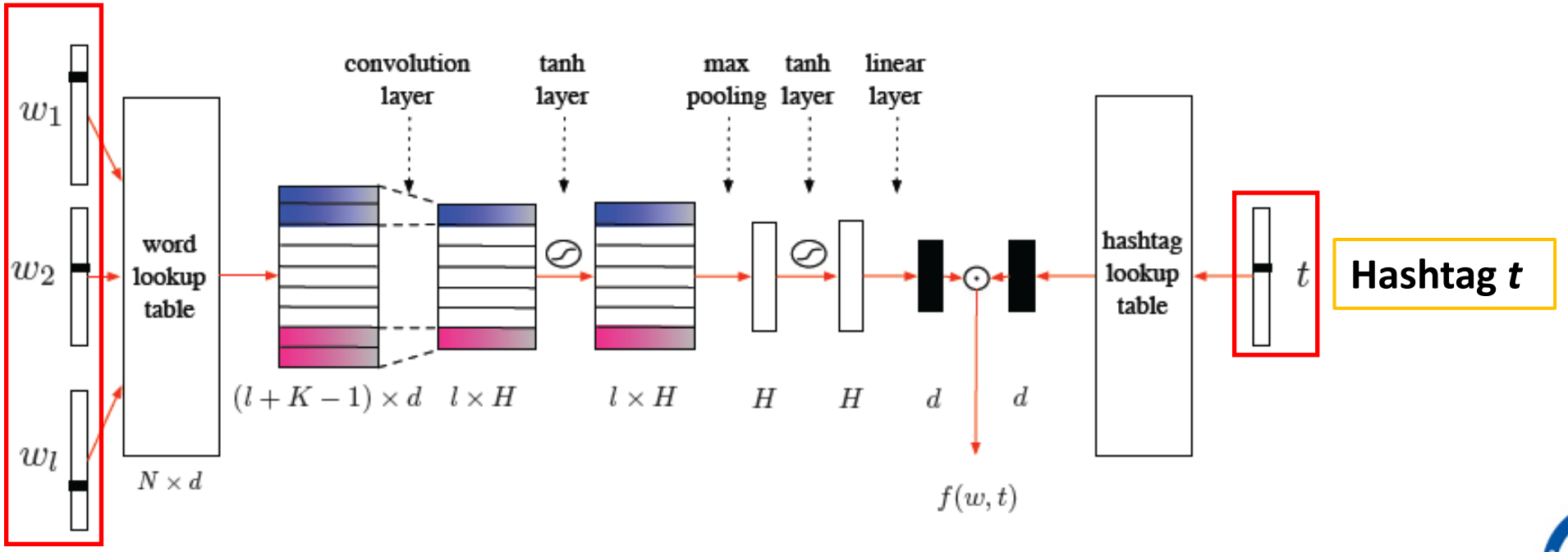
通过池化过程处理，变长特征变为固定全局特征；
基于全部特征继续特征抽取过程。



用户文本理解

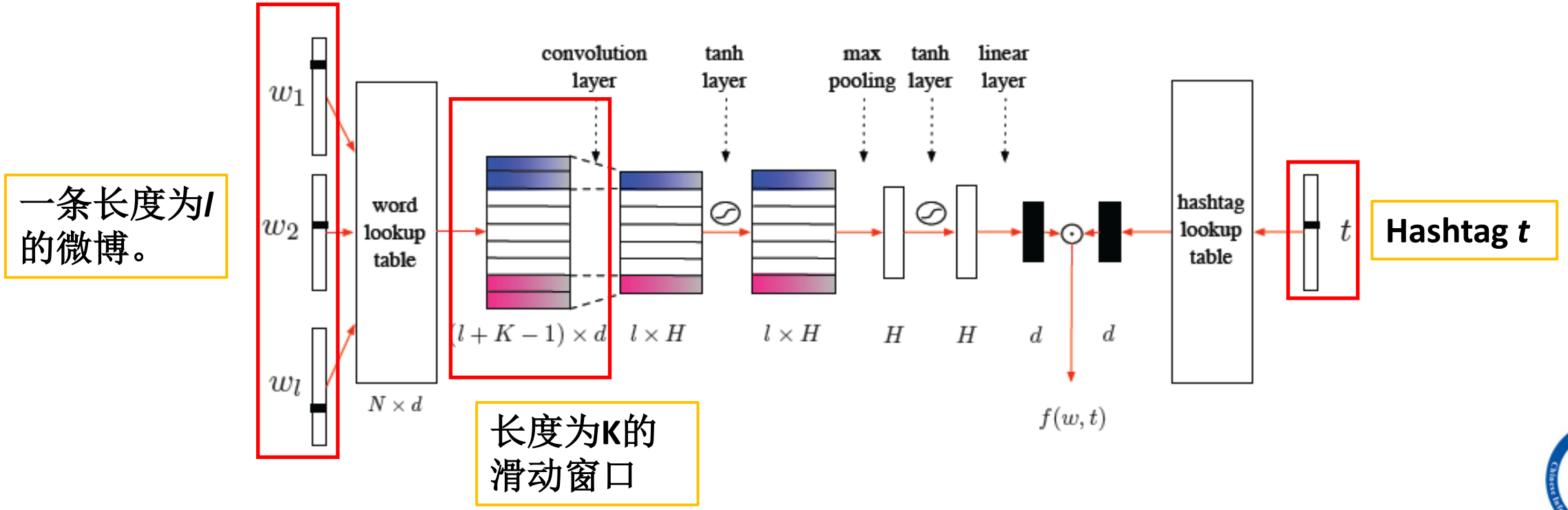
- 借助Hashtag的微博文本语义Embedding [Weston et al EMNLP2014]
  - 运用CNN模型将单词、微博和Hashtag用同一向量空间表示
  - TAGSPACE CNN

一条长度为*l*的微博。



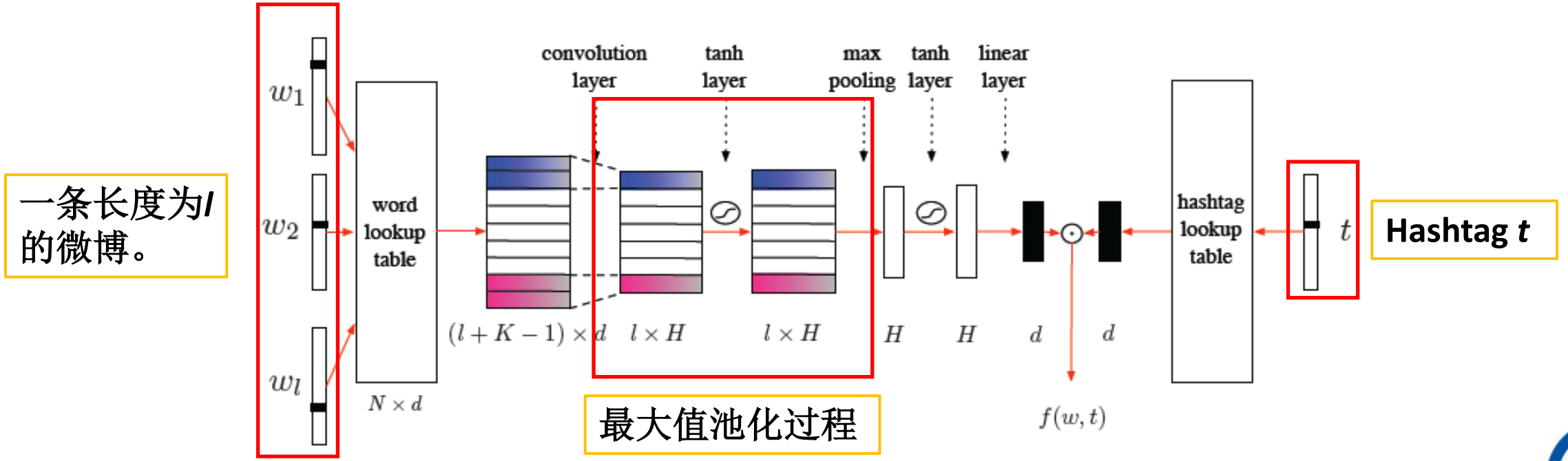
用户文本理解

- 借助Hashtag的微博文本语义Embedding [Weston et al EMNLP2014]
  - 运用CNN模型将单词、微博和Hashtag用同一向量空间表示
  - TAGSPACE CNN



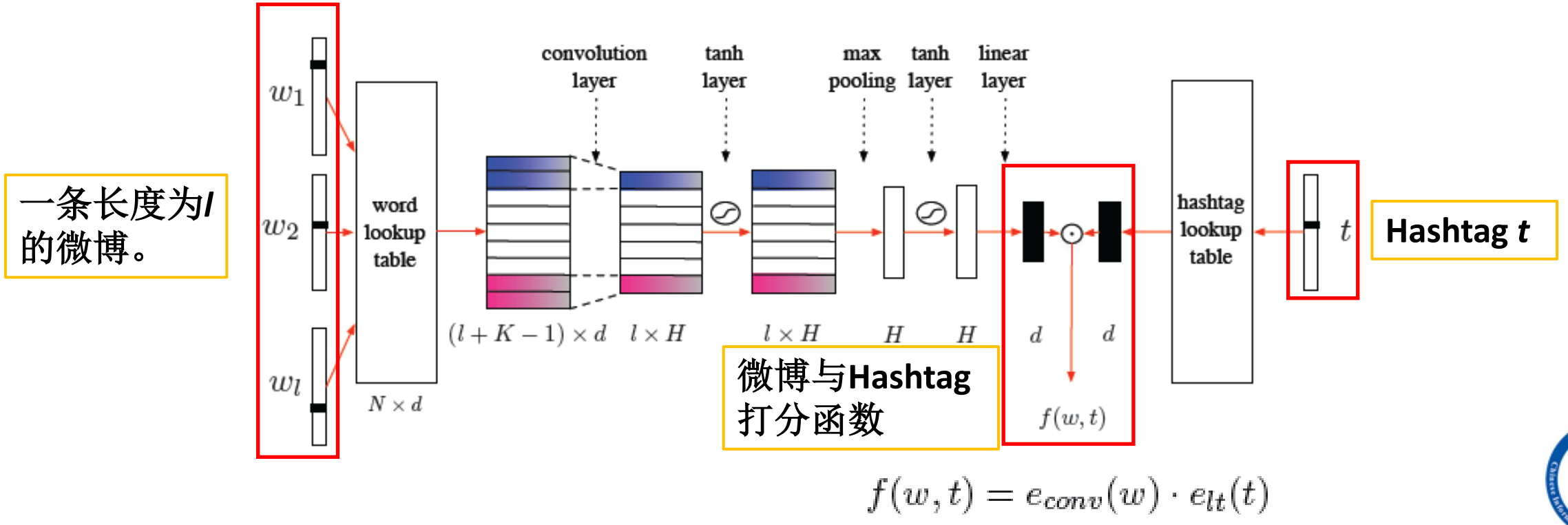
用户文本理解

- 借助Hashtag的微博文本语义Embedding [Yu et al ACL 2016]
  - 运用CNN模型将单词、微博和Hashtag用同一向量空间表示
  - TAGSPACE CNN



用户文本理解

- 借助Hashtag的微博文本语义Embedding [Weston et al EMNLP2014]
  - 运用CNN模型将单词、微博和Hashtag用同一向量空间表示
  - TAGSPACE CNN



用户文本理解

- 借助Hashtag的微博文本语义Embedding [Weston et al EMNLP2014]
  - 运用CNN模型将单词、微博和Hashtag用同一向量空间表示
  - TAGSPACE CNN
  - 模型训练： Hinge Loss + Negative Sampling

$$\mathcal{L} = \max\{0, m - f(w, t^+) + f(w, \bar{t})\}.$$

Margin:
m=1

Hashtag正例

Hashtag负例

用户文本理解

- 实验对比：借助Hashtag的微博文本语义Embedding [Weston et al EMNLP2014]
  - 数据集：

Dataset	Posts (millions)	Words (billions)	Top 4 tags
Pages	35.3	1.6	#fitness, #beauty, #luxury, #cars
People	201	5.5	#FacebookIs10, #love, #tbt, #happy

  - 对比方法：
  - Frequence：基于Hashtag的频率排序
  - #Words: 在Frequence基础上优先考虑出现在微博中的单词（例如：#crazy, #commute, #this）
  - Word2Vec: 运用Word2Vec，将Hashtag当作普通单词；每条微博为单词向量的均值
  - WSABIE：监督型Hashtag与单词向量学习模型，每条微博为单词向量的均值[Weston et al AAAI 2011]

用户文本理解

- 实验对比：借助Hashtag的微博文本语义Embedding [Weston et al EMNLP2014]
- Hashtag Prediction Task

Method	dim	P@1	R@10	Rank
Freq. baseline	-	1.06%	2.48%	11277
#words baseline	-	0.90%	3.01%	11034
Word2Vec	256	1.21%	2.85%	9973
Word2Vec	512	1.14%	2.93%	8727
WSABIE	64	4.55%	8.80%	6921
WSABIE	128	5.25%	9.33%	6208
WSABIE	256	5.66%	10.34%	5519
WSABIE	512	5.92%	10.74%	5452
#TAGSPACE	64	6.69%	12.42%	3569
#TAGSPACE	128	6.91%	12.57%	3858
#TAGSPACE	256	**7.37%**	**12.58%**	**3820**

Table 3: Hashtag test results for people dataset.

Method	dim	P@1	R@10	Rank
Freq. baseline	-	4.20%	1.59%	11103
#words baseline	-	2.63%	5.05%	10581
Word2Vec	256	4.66%	8.15%	10149
Word2Vec	512	5.26%	9.33%	9800
WSABIE	64	24.45%	29.64%	2619
WSABIE	128	27.47%	32.94%	2325
WSABIE	256	29.76%	35.28%	1992
WSABIE	512	30.90%	36.96%	1184
#TAGSPACE	64	34.08%	38.96%	1184
#TAGSPACE	128	36.27%	41.42%	1165
#TAGSPACE	256	**37.42%**	**43.01%**	**1155**

Table 4: Hashtag test results for pages dataset.

监督型方法明显优于非监督型方法

用户文本理解

- 实验对比：借助Hashtag的微博文本语义Embedding [Weston et al EMNLP2014]
- Hashtag Prediction Task

Method	dim	P@1	R@10	Rank
Freq. baseline	-	1.06%	2.48%	11277
#words baseline	-	0.90%	3.01%	11034
Word2Vec	256	1.21%	2.85%	9973
Word2Vec	512	1.14%	2.93%	8727
WSABIE	64	4.55%	8.80%	6921
WSABIE	128	5.25%	9.33%	6208
WSABIE	256	5.66%	10.34%	5519
WSABIE	512	5.92%	10.74%	5452
#TAGSPACE	64	6.69%	12.42%	3569
#TAGSPACE	128	6.91%	12.57%	3858
#TAGSPACE	256	**7.37%**	**12.58%**	**3820**

Table 3: Hashtag test results for people dataset.

Method	dim	P@1	R@10	Rank
Freq. baseline	-	4.20%	1.59%	11103
#words baseline	-	2.63%	5.05%	10581
Word2Vec	256	4.66%	8.15%	10149
Word2Vec	512	5.26%	9.33%	9800
WSABIE	64	24.45%	29.64%	2619
WSABIE	128	27.47%	32.94%	2325
WSABIE	256	29.76%	35.28%	1992
WSABIE	512	30.90%	36.96%	1184
#TAGSPACE	64	34.08%	38.96%	1184
#TAGSPACE	128	36.27%	41.42%	1165
#TAGSPACE	256	**37.42%**	**43.01%**	**1155**

Table 4: Hashtag test results for pages dataset.

CNN模型具有更好的语义理解性能

用户文本理解

- 实验对比：借助Hashtag的微博文本语义Embedding [Weston et al EMNLP2014]
- 微博推荐：基于用户前面10条微博，预测第11条交互微博（阅读、点赞）

Method	dim	P@1	R@10	R@50
Word2Vec	256	0.75%	1.96%	3.82%
BoW	-	1.36%	4.29%	8.03%
WSABIE	64	0.98%	3.14%	6.65%
WSABIE	128	1.02%	3.30%	6.71%
WSABIE	256	1.01%	2.98%	5.99%
WSABIE	512	1.01%	2.76%	5.19%
#TAGSPACE	64	1.27%	4.56%	9.64%
#TAGSPACE	128	1.48%	4.74%	9.96%
#TAGSPACE	256	**1.66%**	**5.29%**	**10.69%**
WSABIE+ BoW	64	1.61%	4.83%	9.00%
#TAGSPACE+ BoW	64	1.80%	5.90%	11.22%
#TAGSPACE+ BoW	256	**1.92%**	**6.15%**	**11.53%**

CNN模型具有更好的语义理解性能

用户文本理解

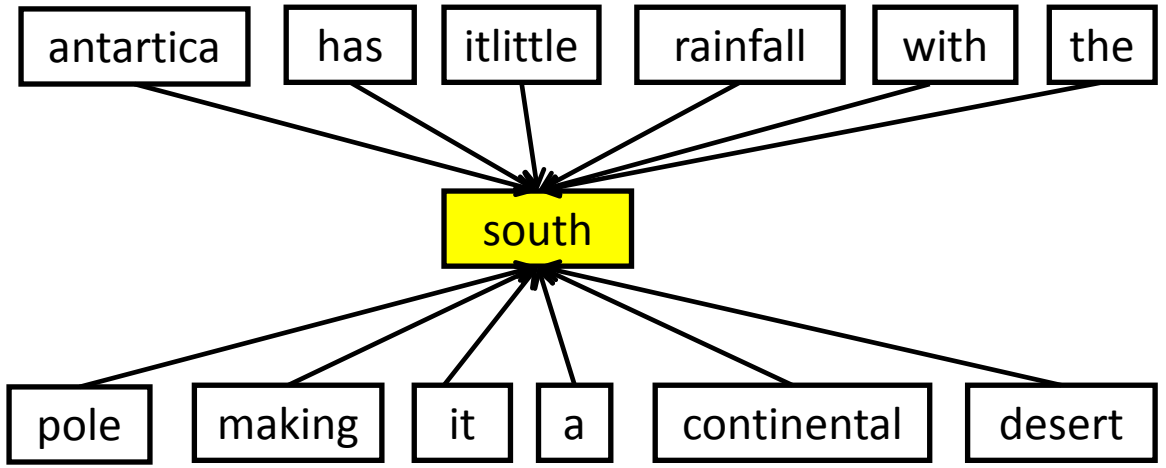
- 微博Hashtag推荐 -- 基于关注机制的CNN模型 [Gong and Zhang IJCAI 2016]
- 注意机制：学习**识别最有用的特征**，并基于**有用特征（权重）**进一步进行**特征抽取**
- 简单来说，就是通过输入特征/中间特征/局部特征学习到一种函数F(X)，

$$F(X) \in R$$

- X可以为输入特征/中间特征/局部特征/，F(X)为学习到的特征的权重，**权重越大**代表特征**更为重要**

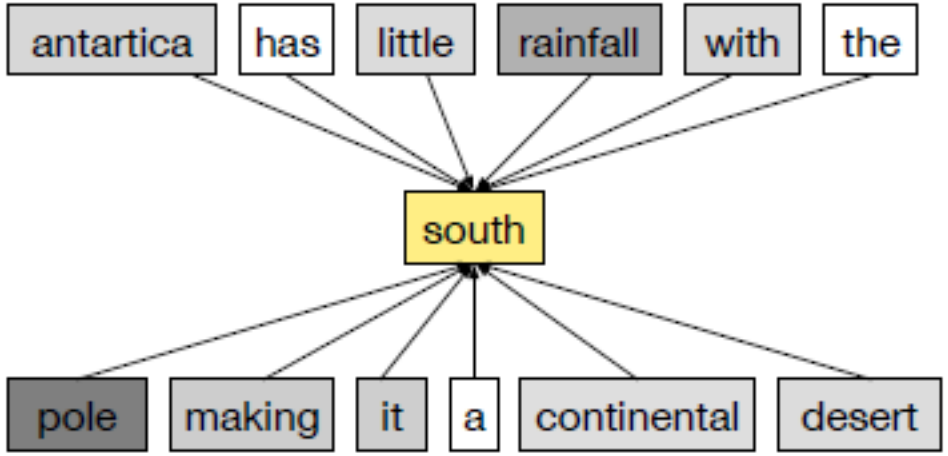
用户文本理解

- 注意机制：学习**识别最有用的特征**，并基于**有用特征（权重）**进一步进行**特征抽取**
- 基于关注机制的单词Embedding技术[Wang et al EMNLP 2015]



CBOW计算单词上下文单词向量的**均值**

语义或语法无关的词带来负面的影响



给予上下文单词向量不同的**权重**

强调语义或语法相关词的作用

用户文本理解

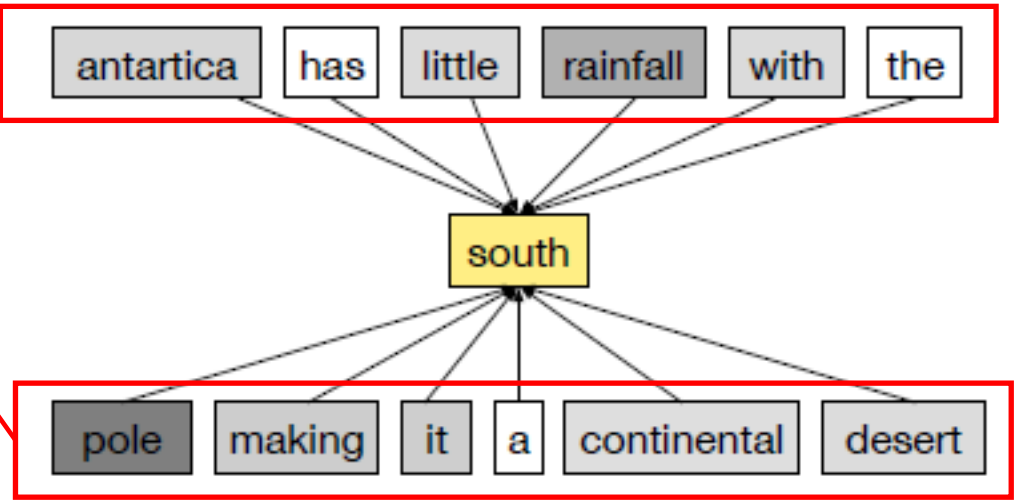
- 注意机制：学习**识别最有用的特征**，并基于**有用特征（权重）**进一步进行**特征抽取**
- 基于关注机制的单词Embedding技术[Wang et al EMNLP 2015]

$$\mathbf{c} = \sum_{i \in [-b,b]-\{0\}} a_i(w_i)\mathbf{w}_i$$

$$a_i(w) = \frac{\exp k_{w,i} + s_i}{\sum_{j \in [-b,b]-\{0\}} \exp k_{w,j} + s_j}$$

单词w在位置i的权重

位置i的偏移量



用户文本理解

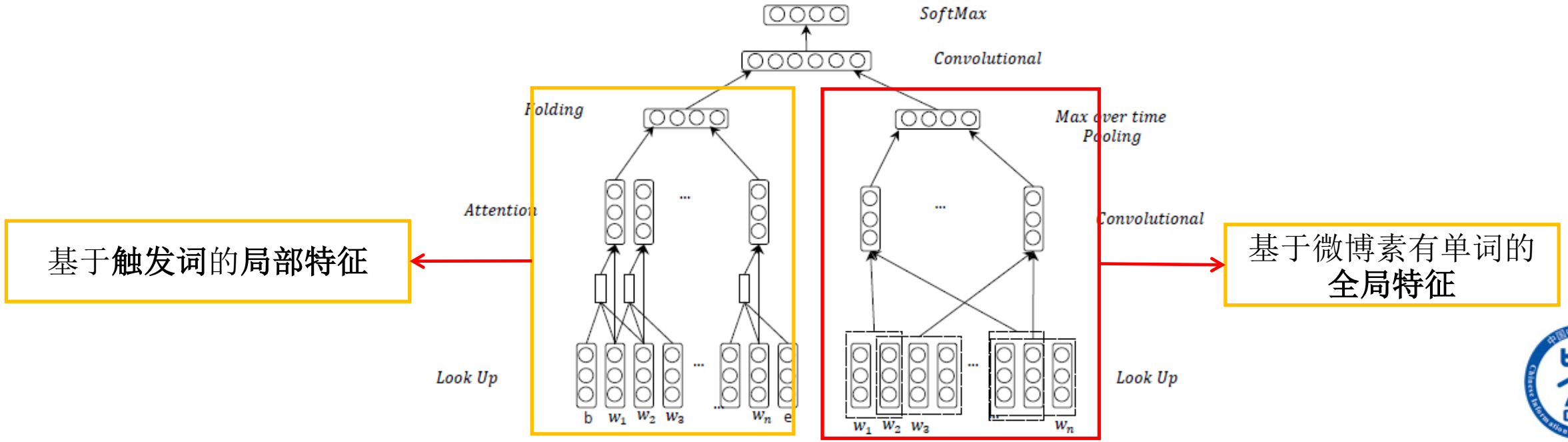
- 注意机制：学习**识别最有用的特征**，并基于**有用特征（权重）**进一步进行**特征抽取**
- 基于注意机制的单词Embedding技术[Wang et al EMNLP 2015]
- 实验对比
  - 基于英文Wikipedia：10M 文章和530M 单词
  - POS标注：基于单词词向量
  - 情感分析：句子用单词词向量的均值表示

在语法分析任务上有显著提高

方法	POS（CoNLL2007 Shared Task）	POS （English PTB）	情感分析（电影评论）
CBOW	50.40	97.03	71.99
Skip-Gram	33.86	97.19	**72.10**
CBOW Attention	**54.00**	**97.39**	71.39

用户文本理解

- 微博Hashtag推荐 -- 基于关注机制的CNN模型 [Gong and Zhang IJCAI 2016]
- 对一条微博抽取局部和全局两类特征
  - 全局特征：基于微博所有单词运用CNN模型抽取特征
  - 局部特征：确定微博中重要语义单词（触发词），基于触发词抽取特征
  - 将全局特征与局部特征结合，通过全连接MLP对每个候选Hashtag打分



用户文本理解

✧ 微博Hashtag推荐 -- 基于关注机制的CNN模型 [Gong and Zhang IJCAI 2016]

基于窗口对中间单词计算权重

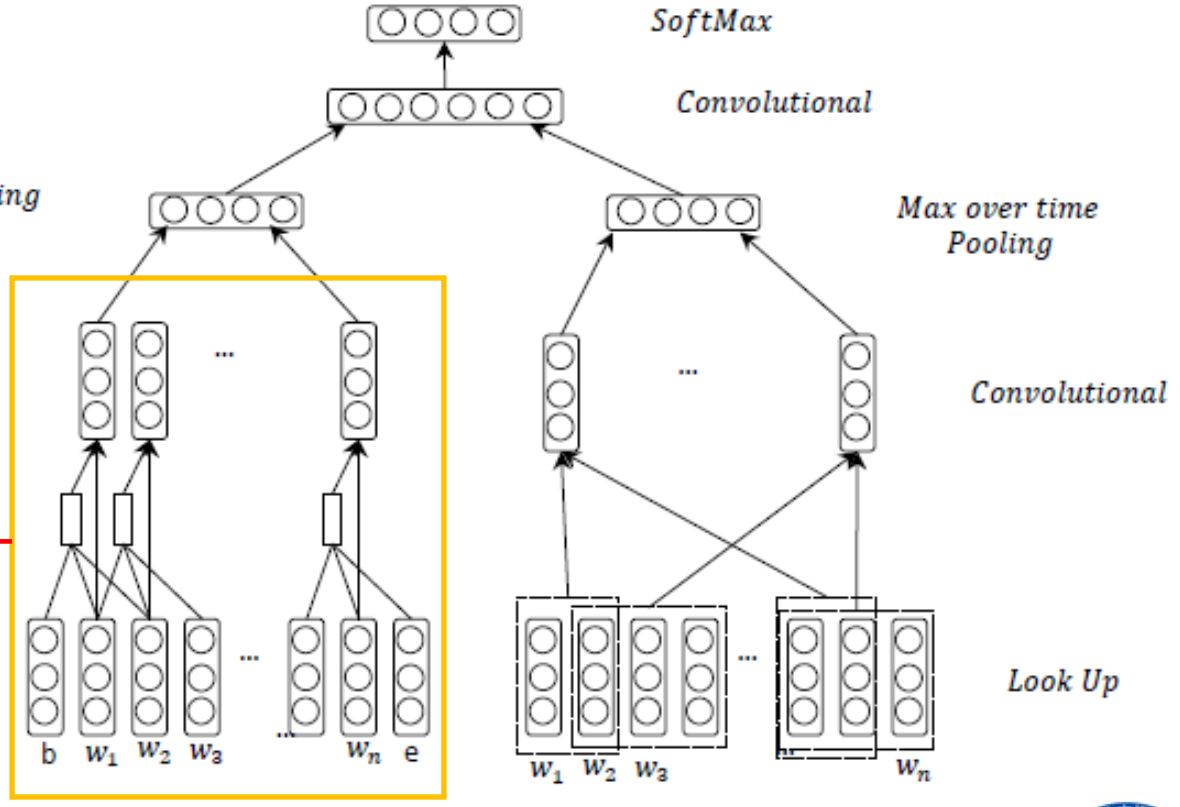
$$s_{(2i+h-1)/2} = g(\mathbf{M}^1 * \mathbf{w}_{i:i+h} + b),$$

$$\widehat{w_i} = \begin{cases} w_i & if \quad s_i > \eta \\ 0 & if \quad s_i \leq \eta \end{cases} \quad 0 \leq i < n,$$

$$\eta = \delta \cdot \min\{\mathbf{s}\} + (1-\delta) \cdot \max\{\mathbf{s}\}$$

线性插值参数

min{s}/max{s}是微博中单词**最小权重**与**最大权重**

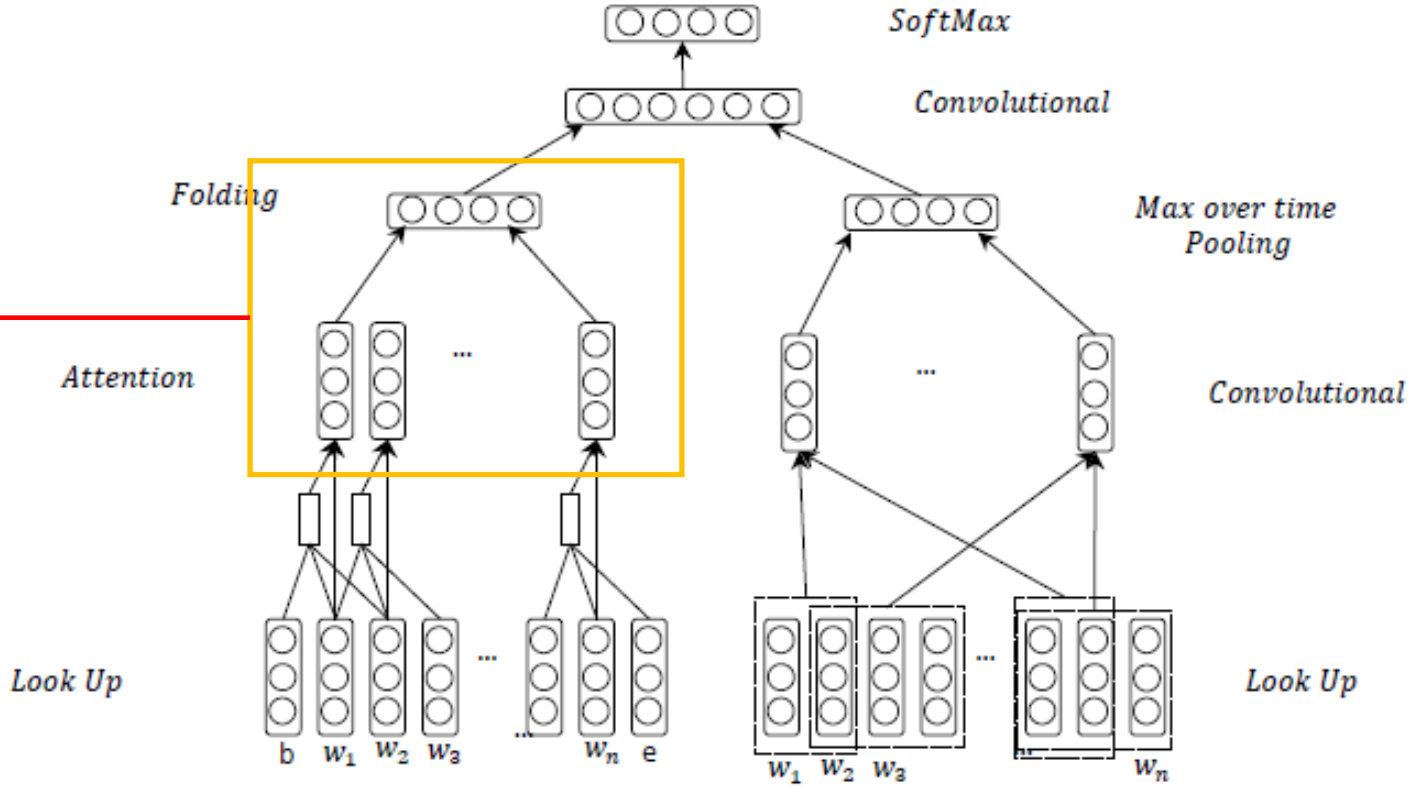


用户文本理解

✧ 微博Hashtag推荐 -- 基于关注机制的CNN模型 [Gong and Zhang IJCAI 2016]

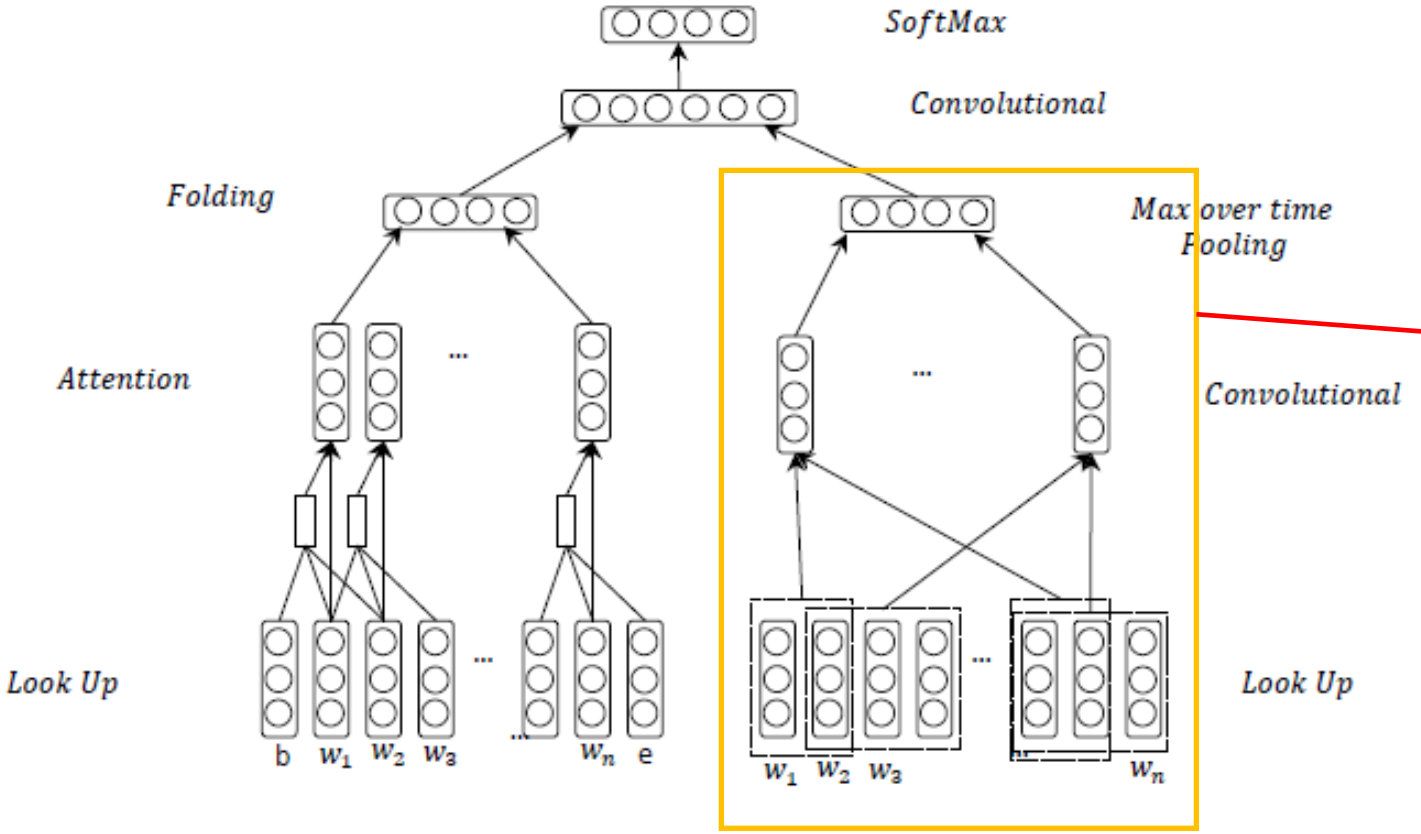
$$z = g(\mathbf{M}^1 * folding(\hat{\mathbf{w}}) + \mathbf{b}),$$

所有触发词的词向量之和



用户文本理解

✧ 微博Hashtag推荐 -- 基于关注机制的CNN模型 [Gong and Zhang IJCAI 2016]



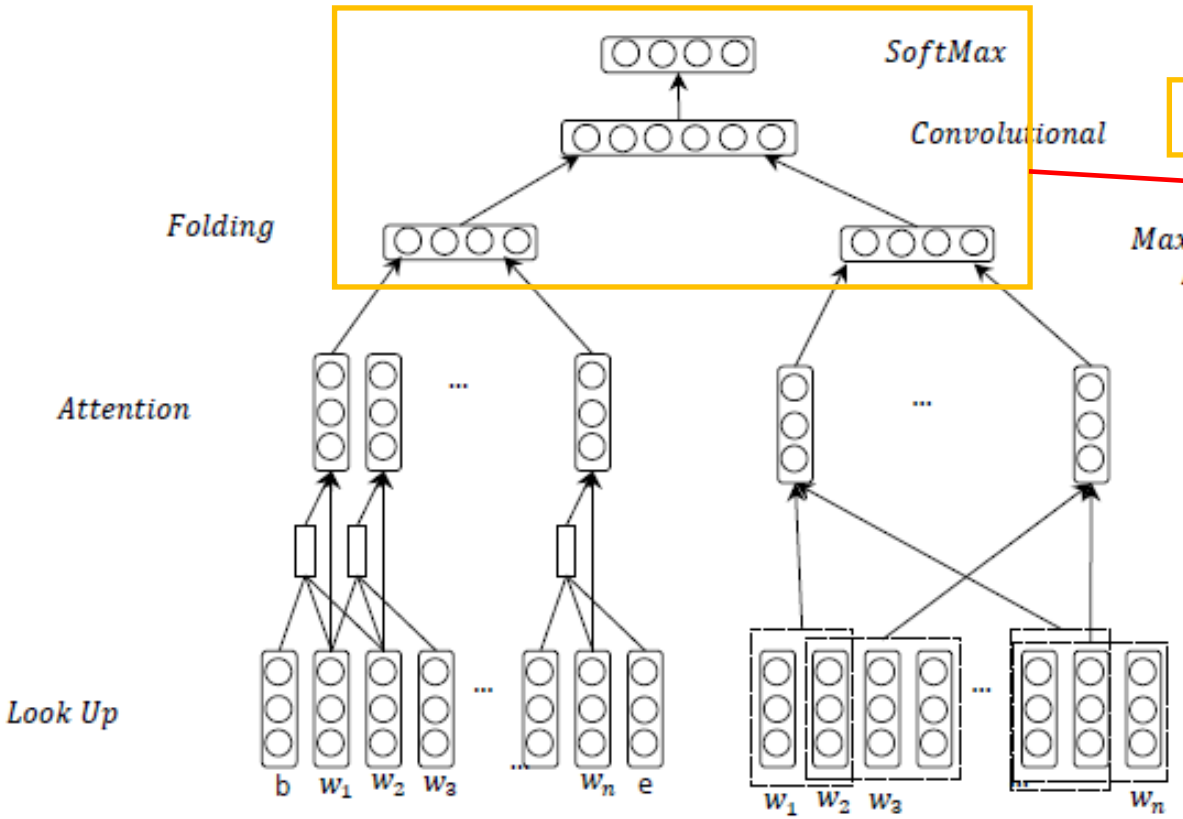
标准CNN过程，池化采用最大值

$$z_i = g(\mathbf{M}^{\mathbf{g}} \cdot \mathbf{w}_{i:i+l-1} + b),$$

窗口大小为l

用户文本理解

✧ 微博Hashtag推荐 -- 基于关注机制的CNN模型 [Gong and Zhang IJCAI 2016]



全连接MLP

$$\hat{\mathbf{h}} = tanh(\mathbf{M} * \mathbf{v}[\mathbf{h_g}; \mathbf{h_l}] + b),$$

触发词的局部特征与全局特征串联

$$P(y^d = a|\mathbf{h}^d; \beta) = \frac{exp(\beta^{(a)T}\mathbf{h}^d)}{\sum_{j \in A} exp(\beta^{(j)T}\mathbf{h}^d)},$$

Hashtag的向量表示

用户文本理解

- 微博Hashtag推荐 -- 基于关注机制的CNN模型 [Gong and Zhang IJCAI 2016]
- 实验数据
  - 新浪微博 110K
  - 词向量预先用Word2Vec进行训练
- 对比方法
  - Naïve Bayes(NB)：基于单词词袋模型的朴素贝叶斯分类器
  - LDA：基于单词共生的主题模型方法 [Krestel et al., RecSys 2009]
  - Translation model (IBM-1)：Hashtag与微博单词的对齐模型 [Liu et al., EMNLP 2011]
  - Topic WA：基于主题的Hashtag与微博单词对齐模型 [Liu et al., COLING 2012]
  - TTM：基于主题的翻译模型 [Ding et al., IJCAI 2013]
  - CNN：基于CNN的分类模型 [Kim et al., arXiv 2014]

用户文本理解

- 微博Hashtag推荐 -- 基于关注机制的CNN模型 [Gong and Zhang IJCAI 2016]
- 性能对比

Methods	Precision	Recall	F_1
NB	0.217	0.197	0.203
LDA	0.064	0.060	0.062
IBM1	0.236	0.214	0.220
TopicWA	0.310	0.285	0.292
TTM	0.382	0.357	0.364
CNN	0.416	0.338	0.373
Attention-5	0.410	0.335	0.369
CNN+Attention-5	**0.443**	**0.362**	**0.398**

传统基于词袋与主题模型的方法比基于CNN的方法性能较差

用户文本理解

- 微博Hashtag推荐 -- 基于关注机制的CNN模型 [Gong and Zhang IJCAI 2016]
- 性能对比

Methods	Precision	Recall	F_1
NB	0.217	0.197	0.203
LDA	0.064	0.060	0.062
IBM1	0.236	0.214	0.220
TopicWA	0.310	0.285	0.292
TTM	0.382	0.357	0.364
CNN	0.416	0.338	0.373
Attention-5	0.410	0.335	0.369
CNN+Attention-5	**0.443**	**0.362**	**0.398**

全局特征CNN模型

基于关注机制的触发词局部特征模型

局部特征与全局特征联合达到最好的效果

用户文本理解

✧ 微博Hashtag推荐 -- 基于关注机制的CNN模型 [Gong and Zhang IJCAI 2016]

基于窗口对中间单词计算权重

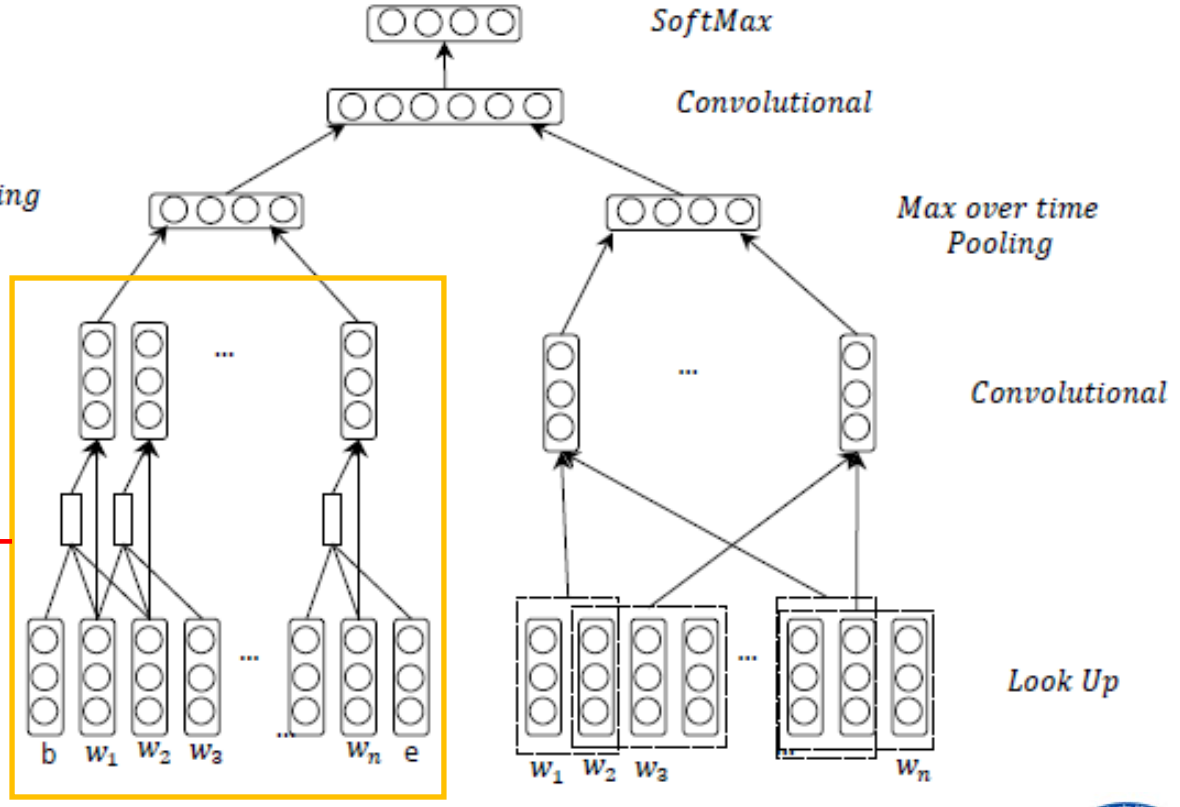
$$s_{(2i+h-1)/2} = g(\mathbf{M}^1 * \mathbf{w}_{i:i+h} + b),$$

$$\widehat{w_i} = \begin{cases} w_i & if \quad s_i > \eta \\ 0 & if \quad s_i \leq \eta \end{cases} \quad 0 \leq i < n,$$

$$\eta = \delta \cdot \min\{\mathbf{s}\} + (1-\delta) \cdot \max\{\mathbf{s}\}$$

线性插值参数

min{s}/max{s}是微博中单词**最小权重**与**最大权重**



用户文本理解

- 微博Hashtag推荐 -- 基于关注机制的CNN模型 [Gong and Zhang IJCAI 2016]
- 性能对比

Methods	Precision	Recall	F_1
Attention-1	0.326	0.267	0.294
Attention-3	0.406	0.332	0.365
Attention-5	0.410	0.335	0.369
Attention-7	0.405	0.331	0.364
CNN-Attention-1	0.395	0.323	0.355
CNN-Attention-3	0.435	0.356	0.392
CNN-Attention-5	0.443	0.362	0.398
CNN-Attention-7	0.438	0.358	0.394

触发词局部特征需要上下文单词作为辅助信息

用户文本理解

- 利用单词Embedding技术开展微博多样化检索 [Onal et al., AIRS 2015]
- 给定一个查询，返回一个包含K条微博的排序，使其最大化排序中每条微博相关度与内容多样性
- 基于现有方法通过单词Embedding计算相识度来扩展查询、微博与查询子话题
  - 最大边缘相关排序 （MMR）

$$f_{MMR}(t_i) = \lambda\, rel(t_i, q) - (1-\lambda) \max_{t_j \in S} sim(t_i, t_j)$$

微博与查询的相关性

微博与已选择微博的最大相关性

用户文本理解

- 利用单词Embedding技术开展微博多样化检索 [Onal et al., AIRS 2015]
- 给定一个查询，返回一个包含K条微博的排序，使其最大化排序中每条微博相关度与内容多样性
- 基于现有方法通过单词Embedding计算相识度来扩展查询、微博与查询子话题
  - 基于查询子话题的多样化排序（xQuAD）

$$f_{xQuAD}(t_i) = (1-\lambda)P(t_i|q) + \lambda \sum_{q_i} \left[P(q_i|q)P(t_i|q_i) \prod_{t_j \in S} (1 - P(t_j|q_i)) \right]$$

微博与查询的相关性

微博满足与子话题的相关性之和

用户文本理解

- 利用单词Embedding技术开展微博多样化检索 [Onal et al., AIRS 2015]
- 基于单词词向量，运用余弦相似度计算单词之间的相似性
- 50D 基于Glove算法的单词词向量
  - 6 billion 单词：Wikipedia & Gigaword
- 对查询、微博进行内容扩展

Resignation	Hillary	Bomb	Budget	Museum
Resign	Rodham	Bombs	Spending	Art
Resigned	Barack	Exploded	Fiscal	Gallery
Appointment	Obama	Explosives	Cuts	Library
Dismissal	Mccain	Blast	Package	Exhibition
Resigning	Clinton	Detonated	Budgets	Museums

$$u_e = u \cup \left(\cup_{i=1}^{m} E(w_i, K)\right)$$

原始查询/微博单词集合u={w_1,w_2,..w_m}

基于单词词向量与w_i最邻近的K个单词

用户文本理解

- 利用单词Embedding技术开展微博多样化检索 [Onal et al., AIRS 2015]
- 查询/微博扩展前后性能比较

Method	a-NDCG		Prec-IA		St-Recall	
	@10	@20	@10	@20	@10	@20
MMR	0.254	0.289	0.042	0.038	0.298	0.438
MMR+Glove	**0.267**	**0.307**	**0.044**	**0.043**	**0.331**	**0.481**
xQuAD	0.325	0.350	0.065	0.053	0.407	0.510
xQuAD+Glove	**0.337**	**0.363**	**0.066**	**0.053**	**0.428**	**0.540**

通过Embedding扩展微博和查询能一定程度缓解短文本带来的语义鸿沟

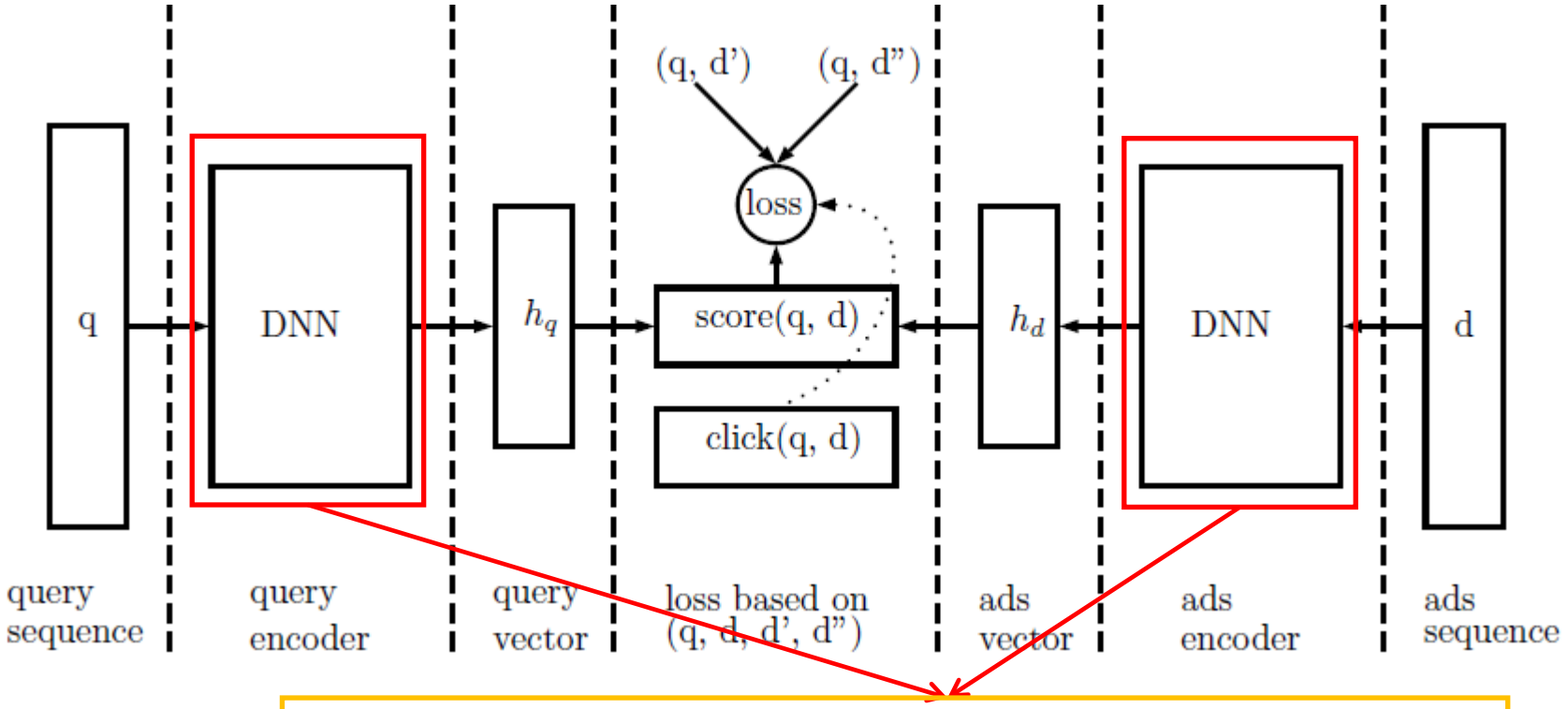
用户文本理解

- 结合RNN和关注机制的在线用户查询理解 [Zhai et al., KDD 2016]
- 通过用户的查询展示与用户查询相关的广告
- 理解用户的查询意图和广告的购买特点
- 现有方法：
  - 基于词袋BOW模型BM25、One-Hot表达
  - PLSA、LDA
- 查询改写
  - 确定包含主要意图的查询单词，去掉无关单词
  - 替换、扩展用户查询

surface | pro | 4 | keyboard

用户文本理解

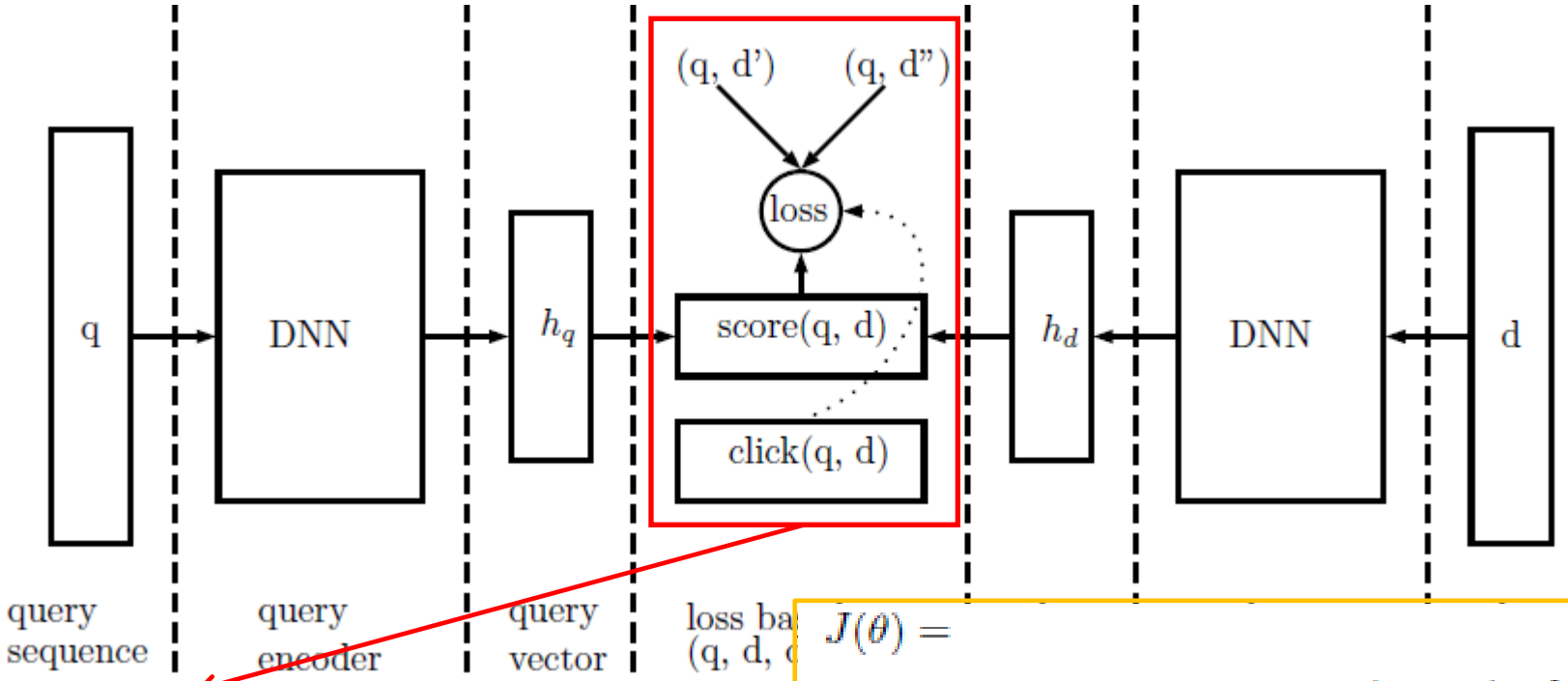
- 结合RNN和关注机制的在线用户查询理解 [Zhai et al., KDD 2016]
- 基于RNN模型抽取用户查询/广告文本的特征表达



基于RNN模型对用户查询语句/广告文本语句进行特征表达
RNN通过对序列建模，更好地捕获语义信息

用户文本理解

- 结合RNN和关注机制的在线用户查询理解 [Zhai et al., KDD 2016]
- 基于RNN模型抽取用户查询/广告文本的特征表达



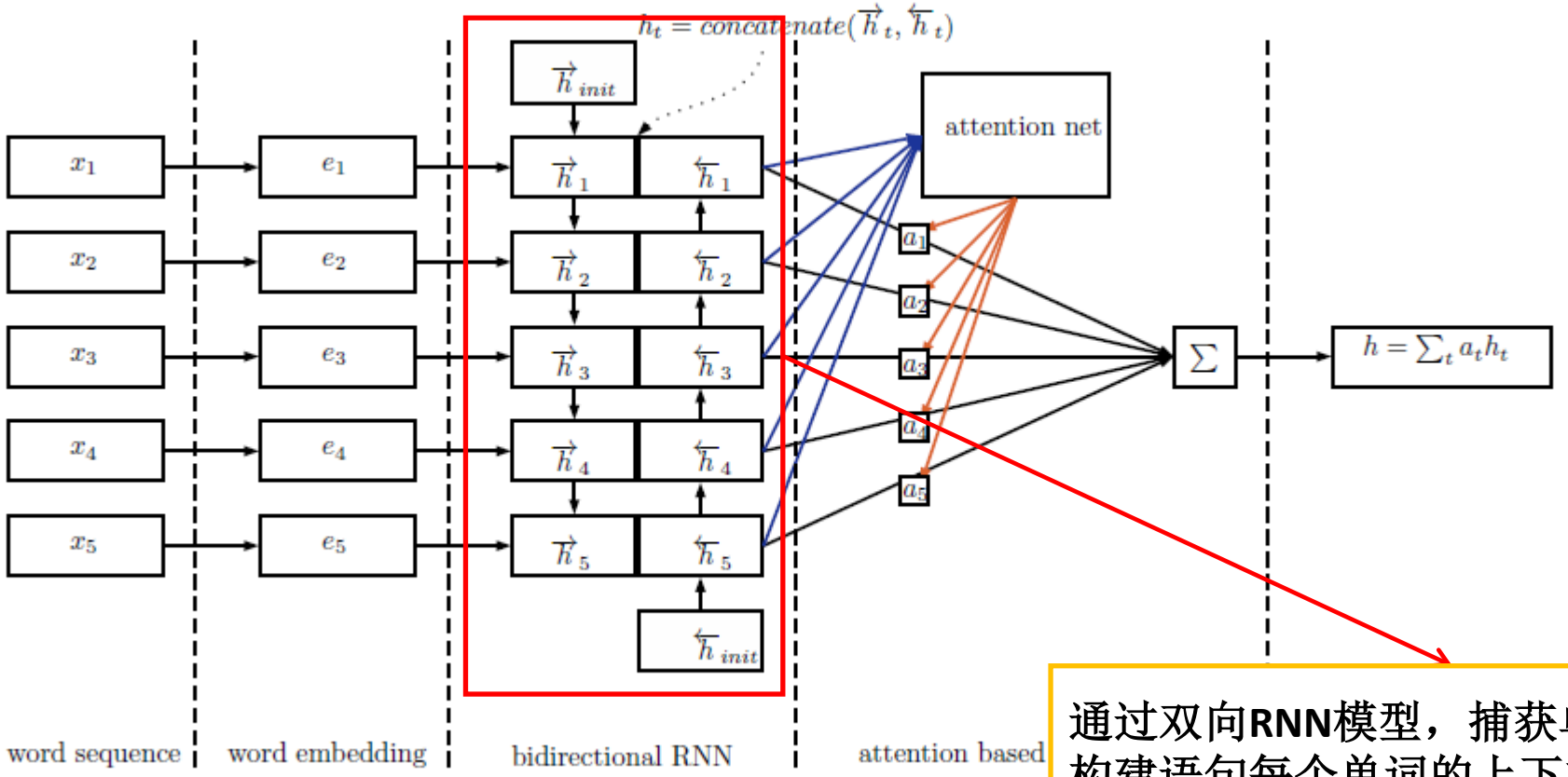
基于负样采样方式，优化目标函数，使用点乘优化正例的分值

$$J(\theta) = -\sum_{(q,d^+)} \log \frac{\exp(score(q,d^+))}{\exp(score(q,d^+)) + \sum_{i=1}^{n} \exp(score(q,d_i^-))}$$

$$s.t.\ score(q,d) = h_q(q)^T h_d(d),$$

用户文本理解

- 结合RNN和关注机制的在线用户查询理解 [Zhai et al., KDD 2016]
- 基于RNN模型抽取用户查询/广告文本的特征表达

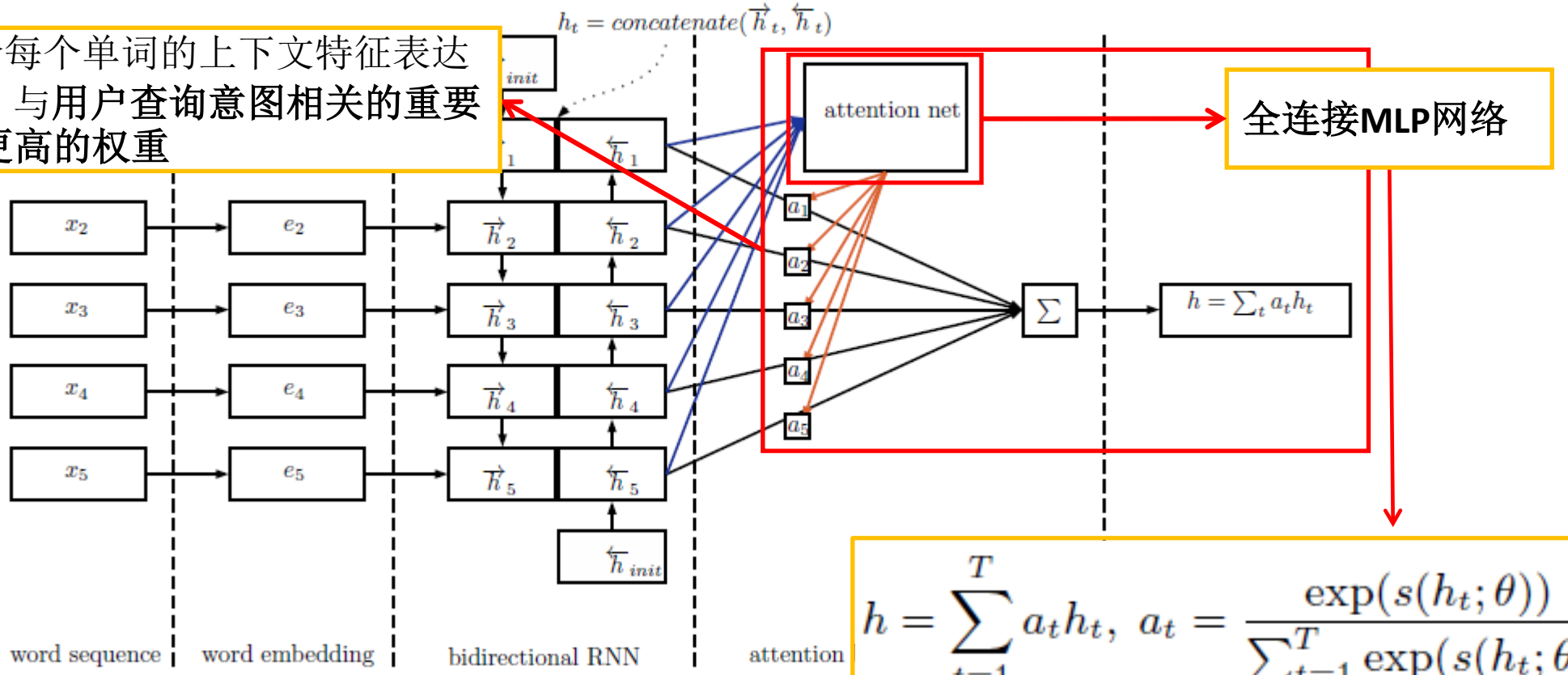


通过双向RNN模型，捕获单词次序信息、构建语句每个单词的上下文特征

用户文本理解

- 结合RNN和关注机制的在线用户查询理解 [Zhai et al., KDD 2016]
- 基于RNN模型抽取用户查询/广告文本的特征表达

运用MLP给每个单词的上下文特征表达计算权重，与**用户查询意图相关的重要单词具有更高的权重**



$$h = \sum_{t=1}^{T} a_t h_t, \; a_t = \frac{\exp(s(h_t;\theta))}{\sum_{t=1}^{T} \exp(s(h_t;\theta))}.$$

用户文本理解

- 结合RNN和关注机制的在线用户查询理解 [Zhai et al., KDD 2016]
- 实验数据：
  - 商用的产品广告搜索引擎
  - 用户查询点击记录 --- 一个月记录、15M点击、6.4M 用户和5.1M广告
  - 测试数据 --- 996K查询商品对 （人工判断相关性）

	size	vocabulary	average length	clicks
query	6.4M	68K	4.1	15M
ads	5.1M	114K	9.3	15M

查询与广告相似性由余弦相似性计算

$$\frac{<h_q(q), h_d(d)>}{\sqrt{\|h_q(q)\|_2^2}\sqrt{\|h_d(d)\|_2^2}}$$

# queries	# ads	# pairs	# positive	# negative
23K	915K	966K	318K	597K

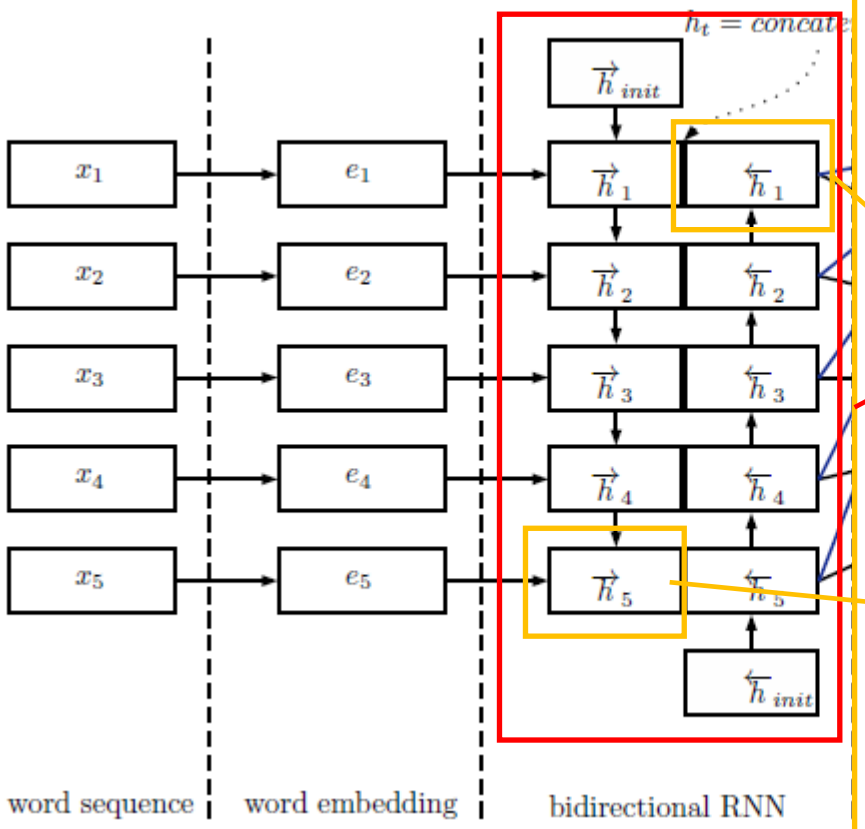
Table 3: Statistics of the testset.

- 实验设置：
  - BoW： $h_t = \sigma(W x_t + b)$
  - RNN、BRNN、LSTM、BLSTM：保留词序信息
  - 特征维度固定为400维

One-Hot表达

用户文本理解

- 结合RNN和关注机制的在线用户查询理解 [Zhai et al., KDD 2016]
- 基于RNN模型抽取用户查询/广告文本的特征表达



采用每个单词的上下文特征的最大值Max Pooling

关注机制的替代选择

基于序列最后单词的特征Last Pooling

用户文本理解

- 结合RNN和关注机制的在线用户查询理解 [Zhai et al., KDD 2016]
- 实验对比

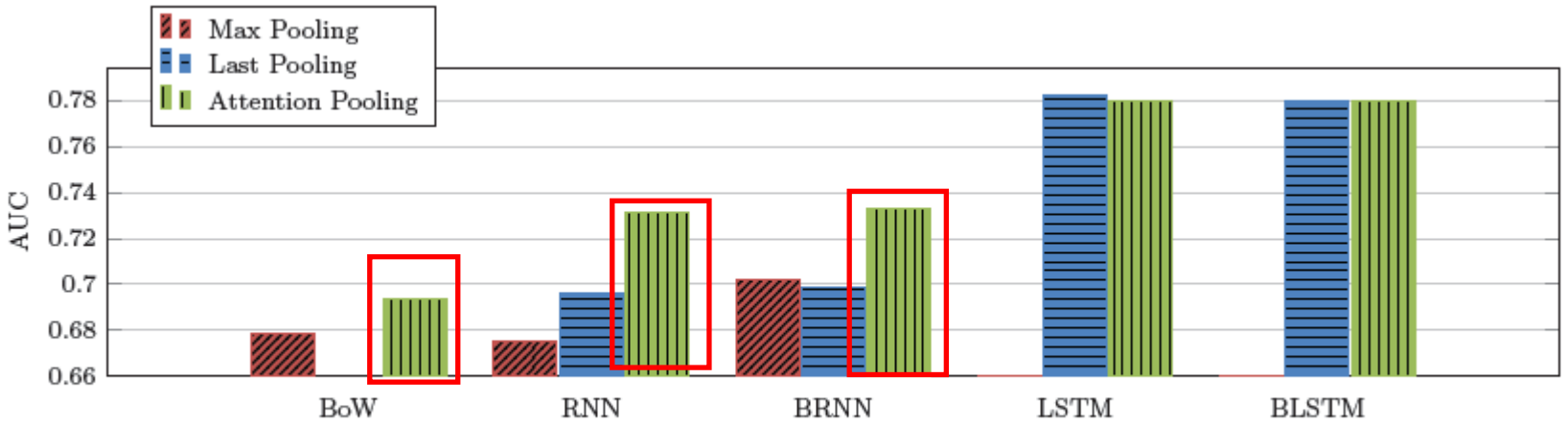


Figure 3: AUC evaluated on the test set of different models.

RNN和BOW模型都收益于关注机制！

用户文本理解

- 结合RNN和关注机制的在线用户查询理解 [Zhai et al., KDD 2016]
- 实验对比

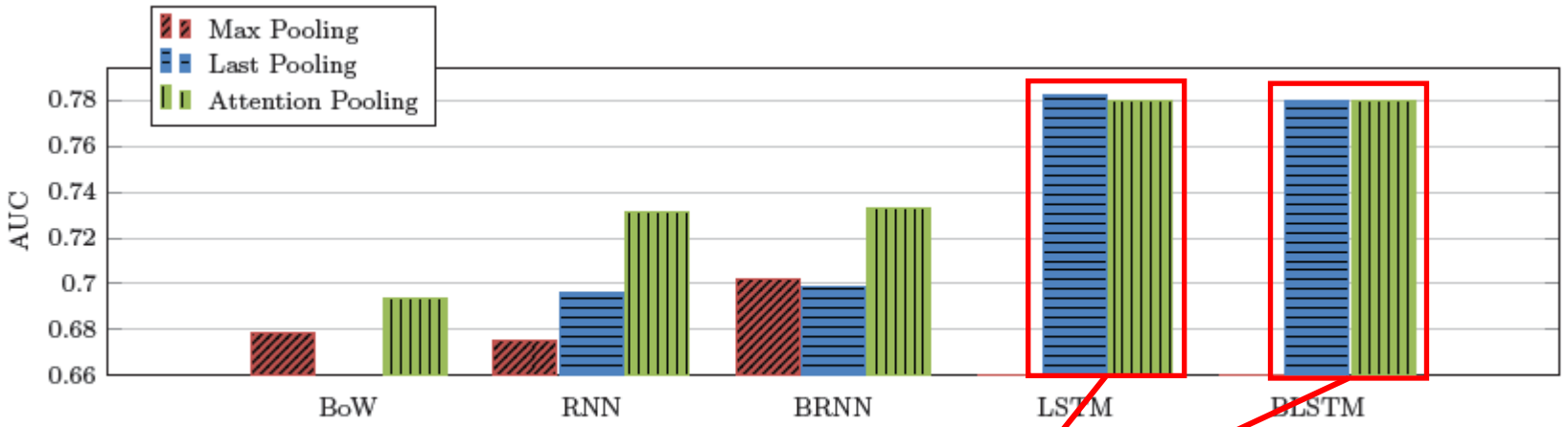


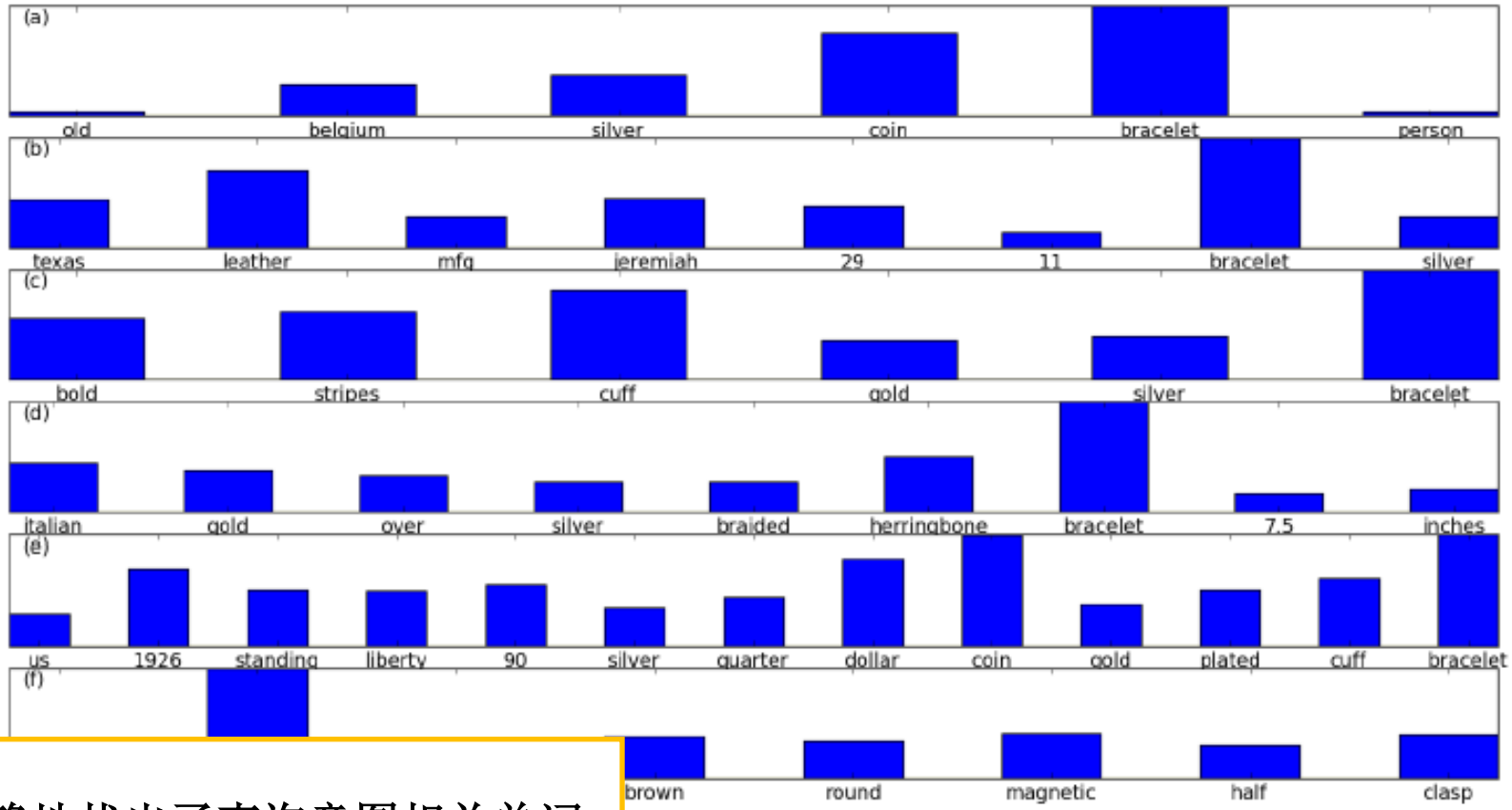
Figure 3: AUC evaluated on the test set of different models.

基于LSTM的抽取模型具有更好的语义理解性能

LSTM可以捕捉长距离的有用信息，因此关注机制作用较少！

用户文本理解

- 结合RNN和关注机制的在线用户查询理解 [Zhai et al., KDD 2016]
- 实验对比



基于BRNN的关注机制准确地找出了查询意图相关单词

Figure 4: Attention score visualization of a query (a) and 5 top ranked ads (b)(c)(d)(e)(f) by BRNN.

用户文本理解

- 结合RNN和关注机制的在线用户查询理解 [Zhai et al., KDD 2016]
- 利用关注机制确定查询单词的权重，基于BM25模型进行查询与相关广告的匹配

$$score(q,d) = \sum_{i=1}^{n} IDF(q_i) \frac{TF(q_i,d)(k_1+1)}{TF(q_i,d) + k_1(1-b+b\frac{|d|}{avgdl})},$$

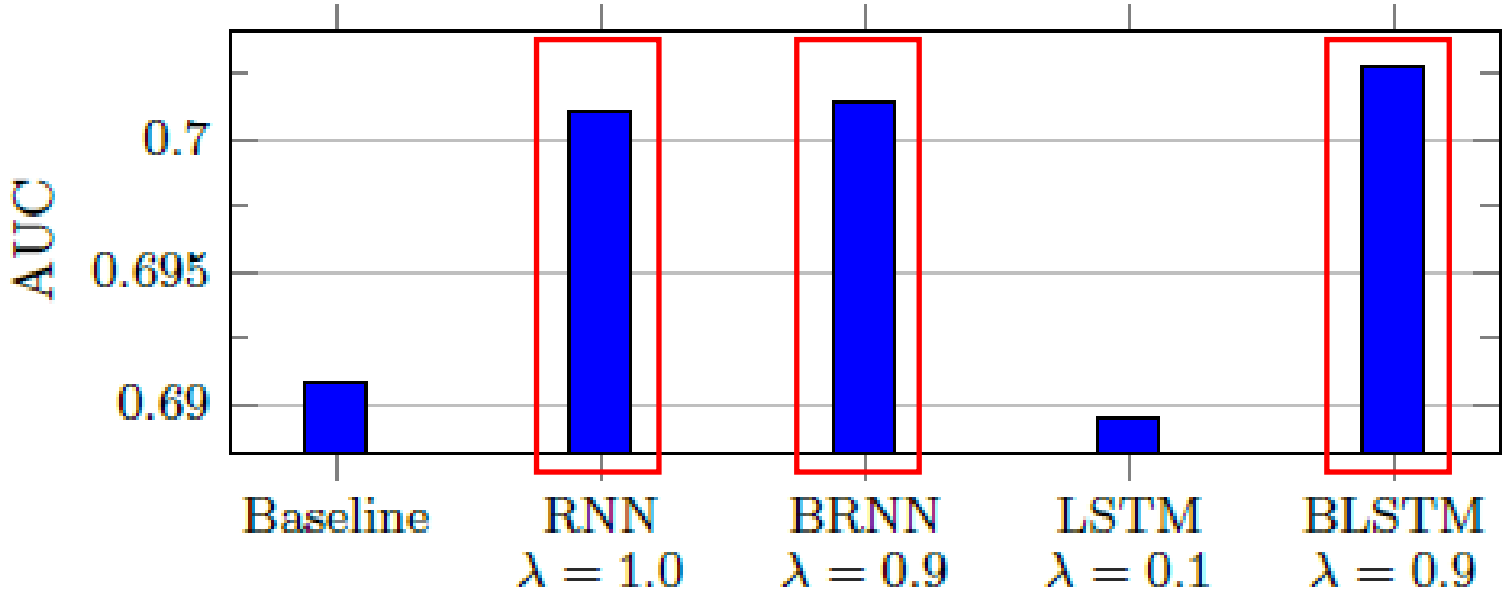
$$TF'(q_i,d) = \sum_{j:d_j=q_i} \lambda \cdot a_j |d| + (1-\lambda)\cdot 1$$

平滑参数，避免TF权重为0.

单词j在检索文档中的基于关注机制的权重

用户文本理解

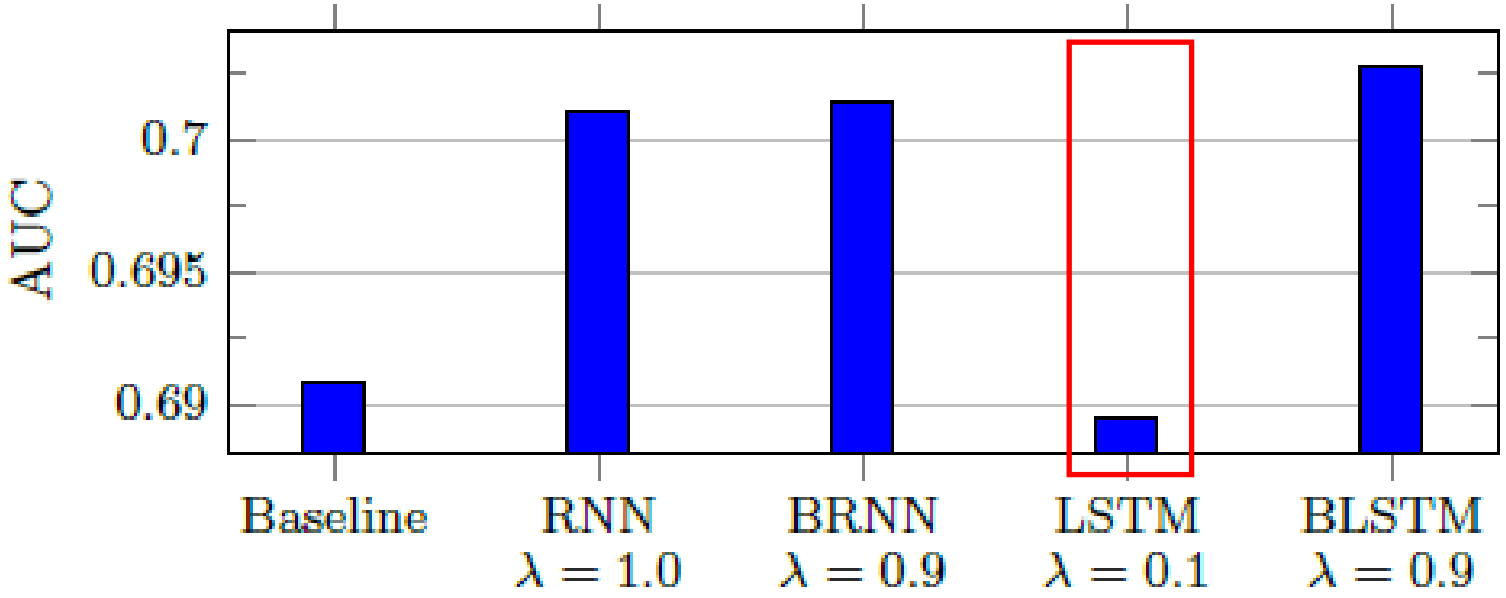
- 结合RNN和关注机制的在线用户查询理解 [Zhai et al., KDD 2016]
- 利用关注机制确定查询单词的权重，基于BM25模型进行查询与相关广告的匹配



基于RNN、BRNN、BLSTM的关注机制学习到的单词权重质量很高

用户文本理解

- 结合RNN和关注机制的在线用户查询理解 [Zhai et al., KDD 2016]
- 利用关注机制确定查询单词的权重，基于BM25模型进行查询与相关广告的匹配

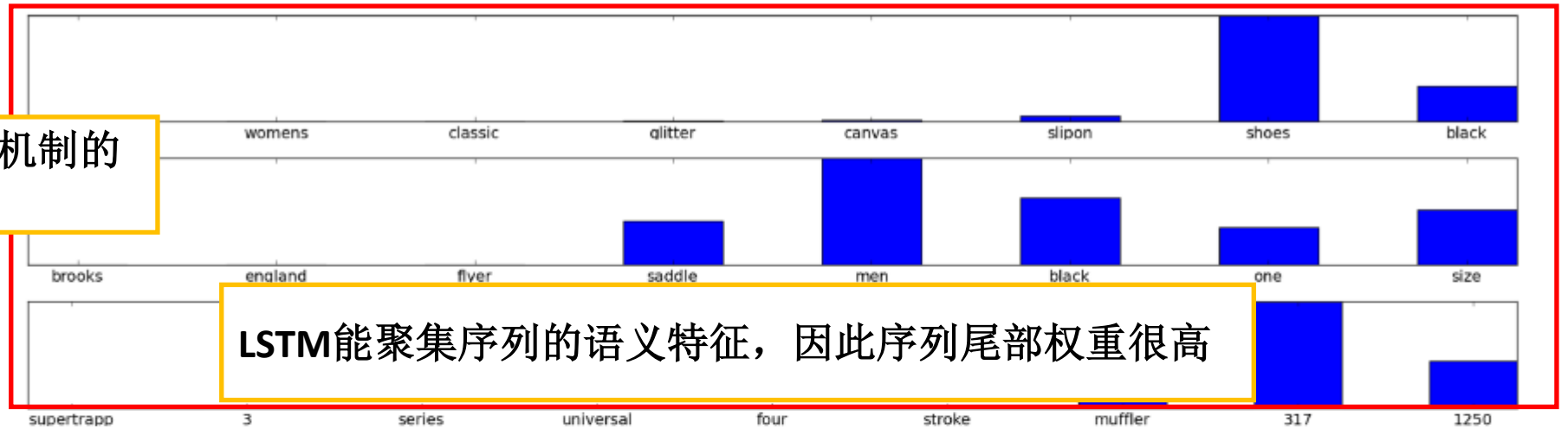


平滑参数接近0， 表明基于LSTM抽取到的特征的关注机制基本失效。

用户文本理解

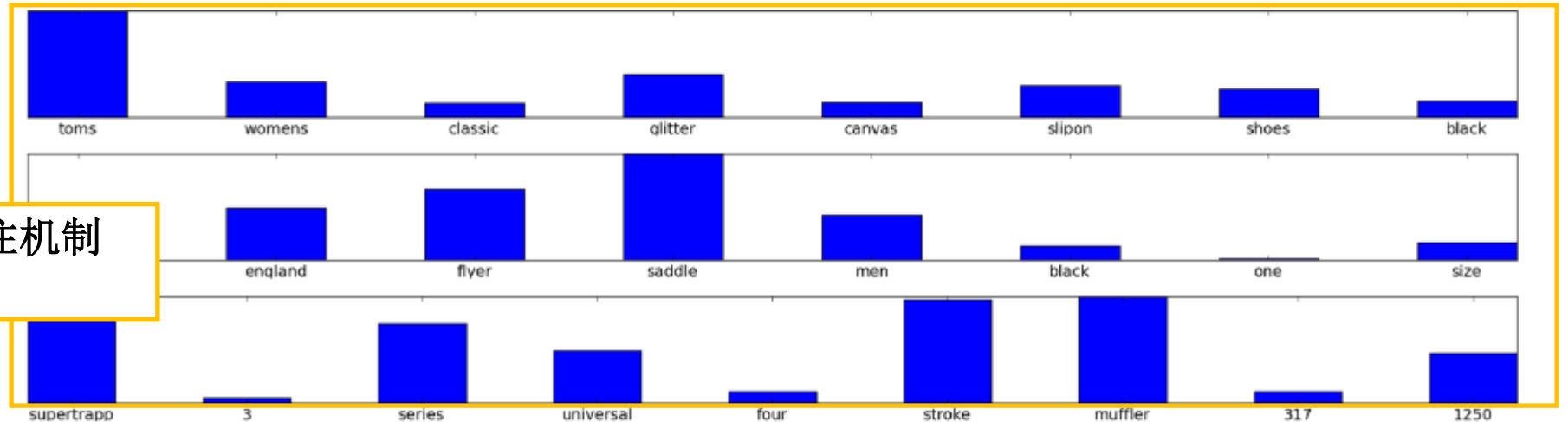
- 结合RNN和关注机制的在线用户查询理解 [Zhai et al., KDD 2016]
- 利用关注机制确定查询单词的权重，基于BM25模型进行查询与相关广告的匹配

基于LSTM的关注机制的权重分布



LSTM能聚集序列的语义特征，因此序列尾部权重很高

基于BLSTM的关注机制的权重分布



用户文本理解

- 基于深度学习的人机对话系统 [Yan et al., SIGIR 2016]
- 通过深度学习理解用户语言语义信息
- 从大量文本问答对中匹配最相关的回答

Posting: 近视了需要戴眼镜... (I need a pair of glasses because of the myopia...)
Reply$_1$: 我送你眼镜！ (I will offer the glasses for you!) *Reply*$_2$: 可以恢复的，别紧张... (You will be recovered. Don't worry.)

用于训练和检索的文本问答对

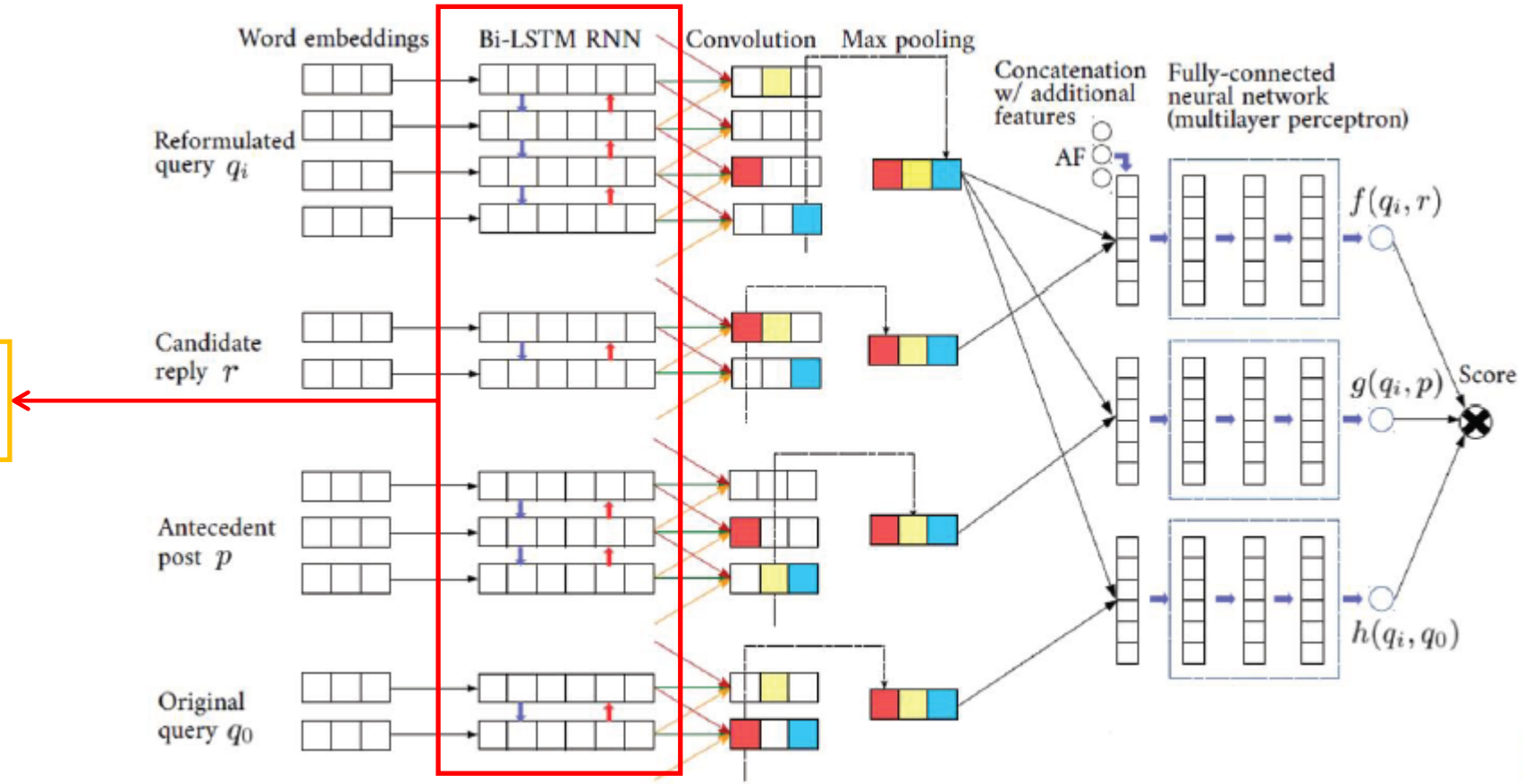
Human-Computer Conversation
A_1: 天哪一把年纪的人居然近视了 (OMG I got myopia at such an "old" age) B_1: 真的吗？ (Really?) A_2:嗯哪。求个眼镜做礼物！ (Yeah. Wish a pair of glasses as a gift.)
B_2:我送你眼镜！ (I will offer the glasses for you!)

通过深度学习网络检索现有最佳回答

用户文本理解

- 基于深度学习的人机对话系统 [Yan et al., SIGIR 2016]
- 运用双向LSTM和CNN叠加模型给用户对话与候选回答进行打分

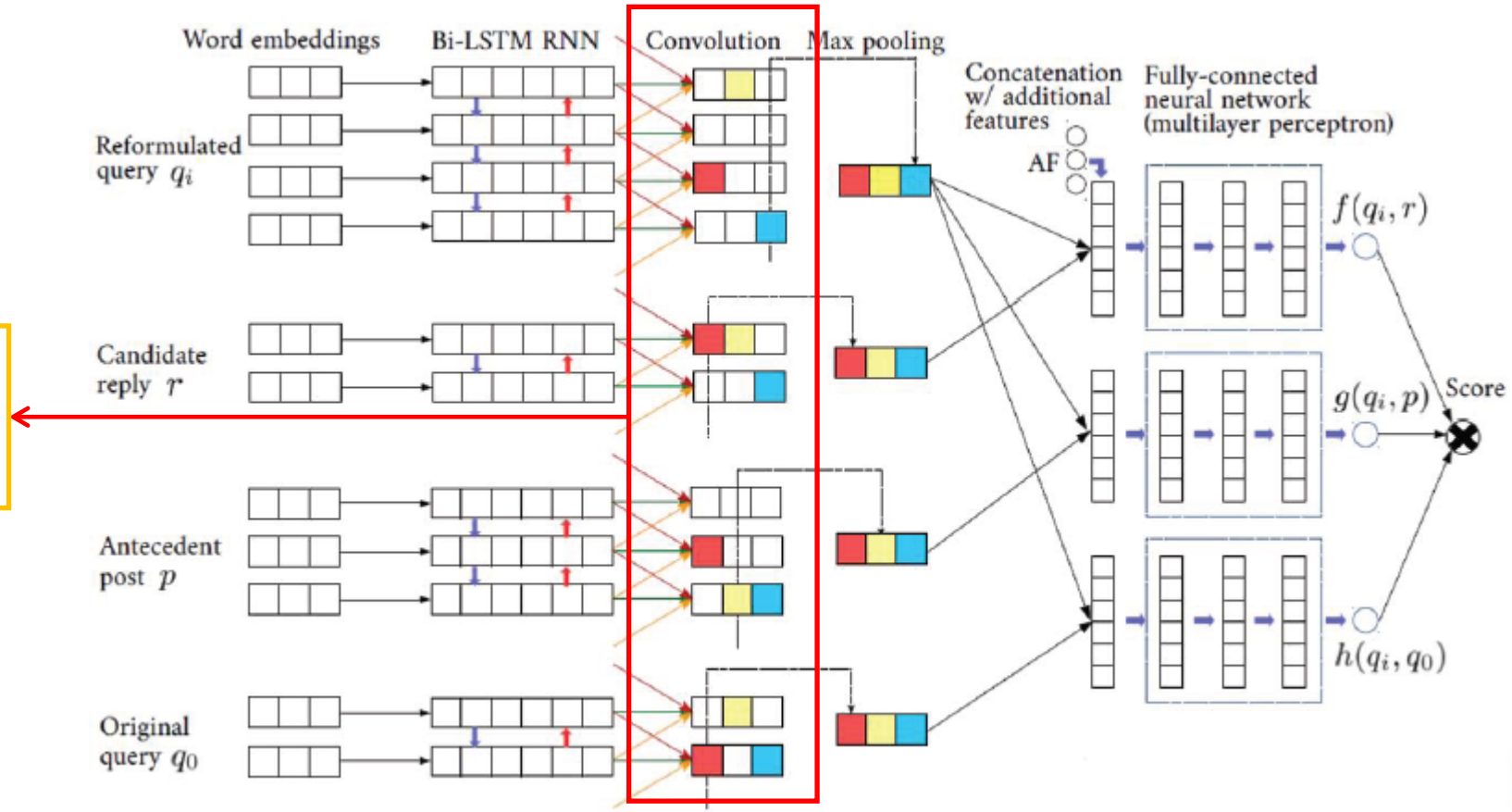
运用双向LSTM抽取每个单词的上下文语义特征



用户文本理解

- 基于深度学习的人机对话系统 [Yan et al., SIGIR 2016]
- 运用双向LSTM和CNN叠加模型给用户对话与候选回答进行打分

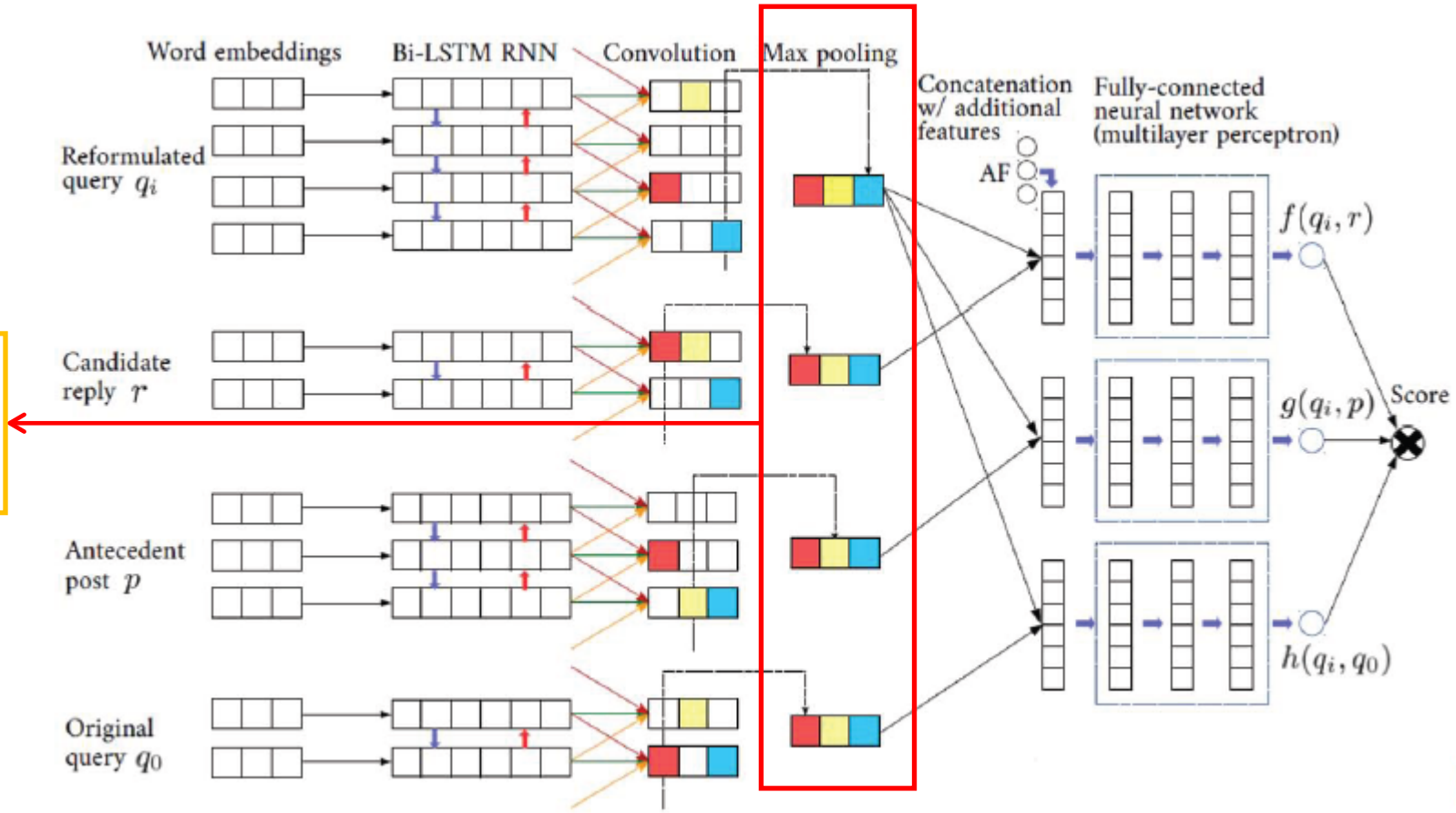
对用户提问与候选回答的每个单词的BLSTM特征，运用CNN模型+最大值池化



用户文本理解

- 基于深度学习的人机对话系统 [Yan et al., SIGIR 2016]
- 运用双向LSTM和CNN叠加模型给用户对话与候选回答进行打分

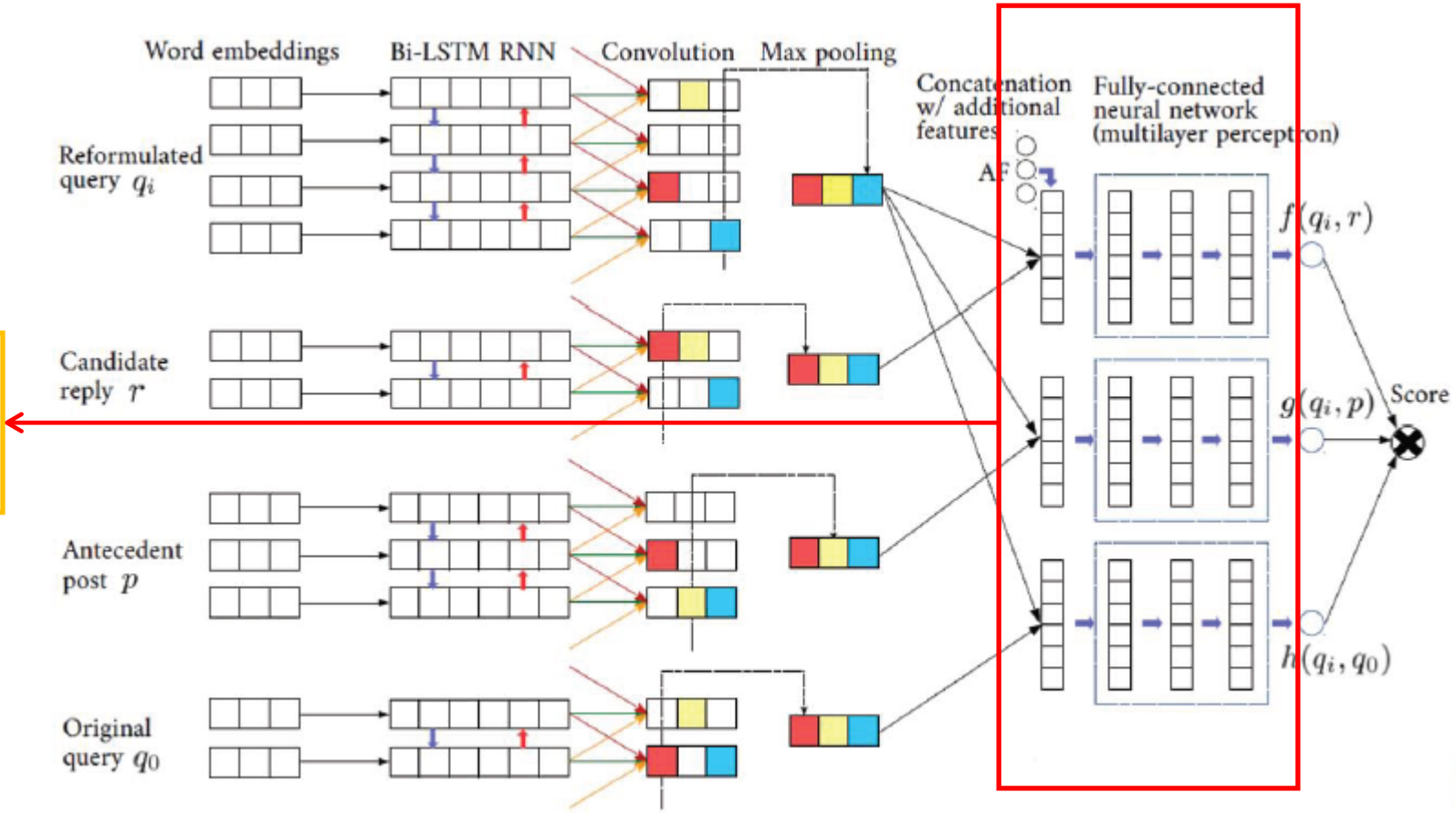
基于CNN的卷积结果，运用最大值池化抽取**固定长度的特征**



用户文本理解

- 基于深度学习的人机对话系统 [Yan et al., SIGIR 2016]
- 运用双向LSTM和CNN叠加模型给用户对话与候选回答进行打分

基于用户提问与候选回答的特征，运用多层全连接网络进行相关度匹配



用户文本理解

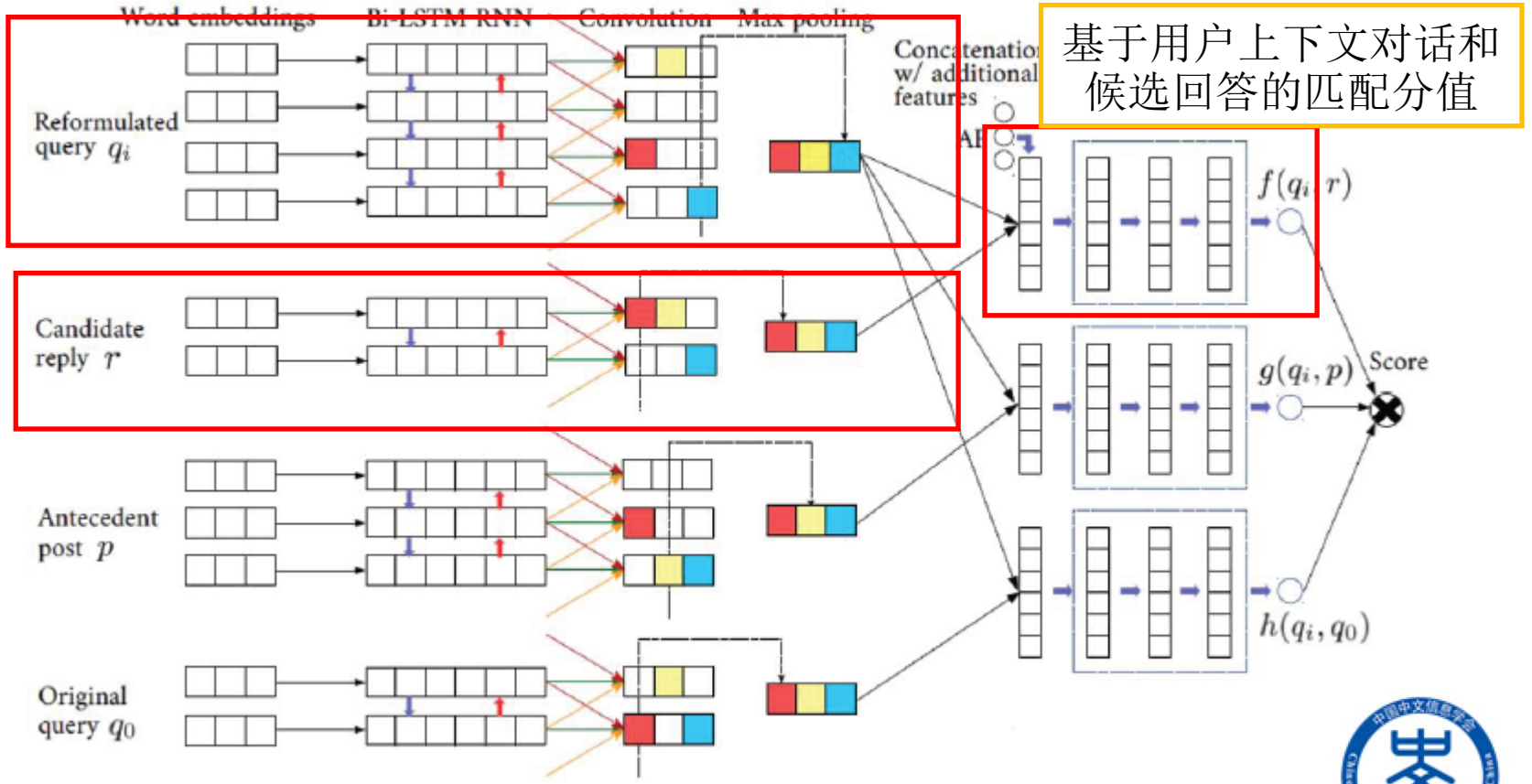
- 基于深度学习的人机对话系统 [Yan et al., SIGIR 2016]
- 通过借助对话上下文信息帮助理解**用户对话意图**

Posting: 近视了需要戴眼镜...
(I need a pair of glasses because of the myopia...)
Reply$_1$: 我送你眼镜！
(I will offer the glasses for you!)
Reply$_2$: 可以恢复的，别紧张...
(You will be recovered. Don't worry.)

原始训练数据/候选回答

Human-Computer Conversation
A_1: 天哪一把年纪的人居然近视了
(OMG I got myopia at su...)
B_1: 真的吗？
(Really?)
A_2:嗯哪。求个眼镜做礼物！
(Yeah. Wish a pair of glasses as a gift.)
B_2:我送你眼镜！
(I will offer the glasses for you!)

用户上下文对话内容



基于用户上下文对话和候选回答的匹配分值

用户文本理解

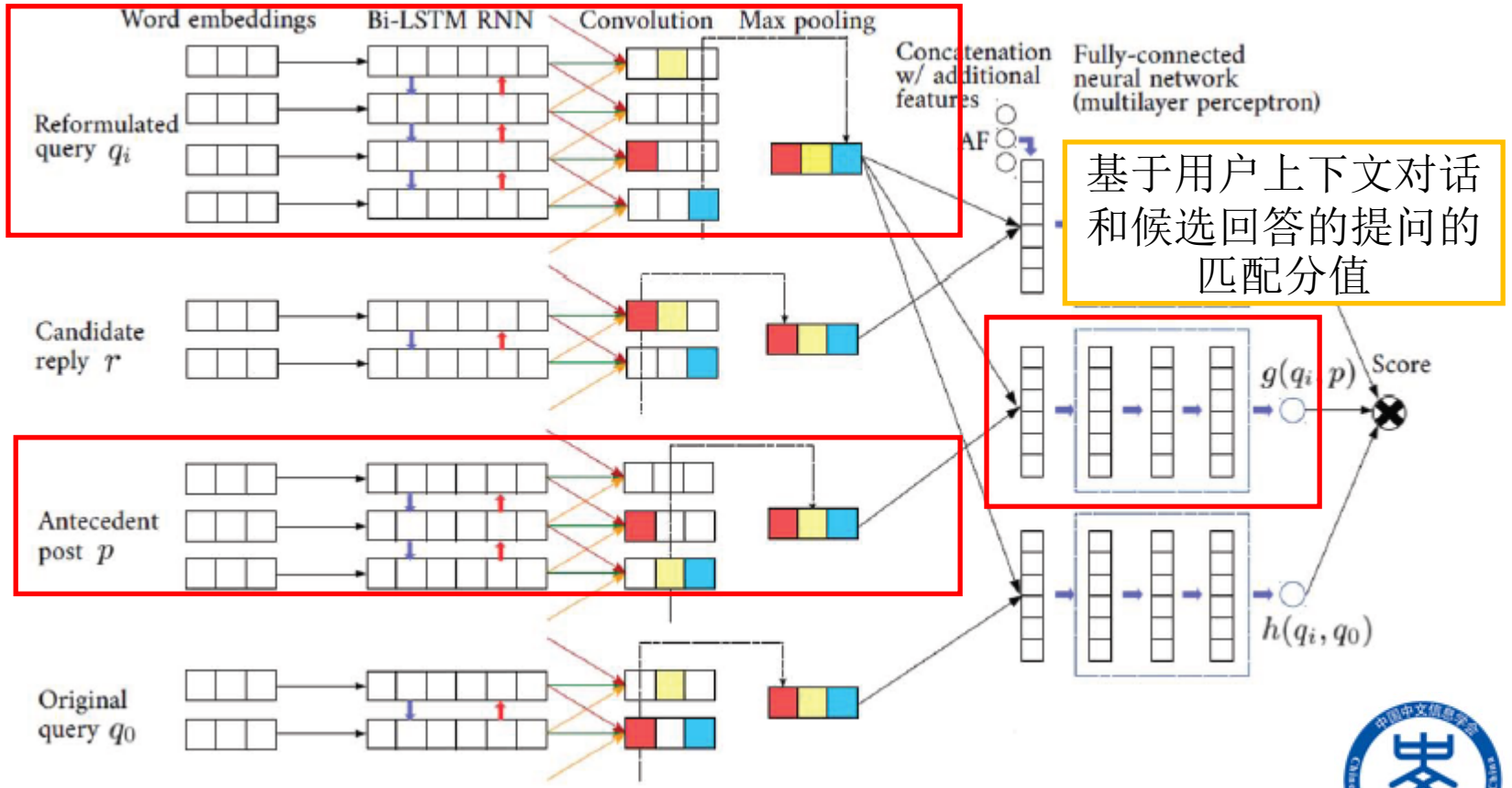
- 基于深度学习的人机对话系统 [Yan et al., SIGIR 2016]
- 通过借助对话上下文信息帮助理解**用户对话意图**

Posting: 近视了需要戴眼镜...
(I need a pair of glasses because of the myopia...)
Reply$_1$: 我送你眼镜
(I will offer the gla
Reply$_2$: 可以恢复的
(You will be recovered. Don't worry.)

原始训练数据/候选回答的提问

Human-Computer Conversation
A_1: 天哪一把年纪的人居然近视了
(OMG I got myopia at su
B_1: 真的吗？
(Really?)
A_2:嗯哪。求个眼镜做礼物！
(Yeah. Wish a pair of glasses as a gift.)
B_2:我送你眼镜！
(I will offer the glasses for you!)

用户上下文对话内容

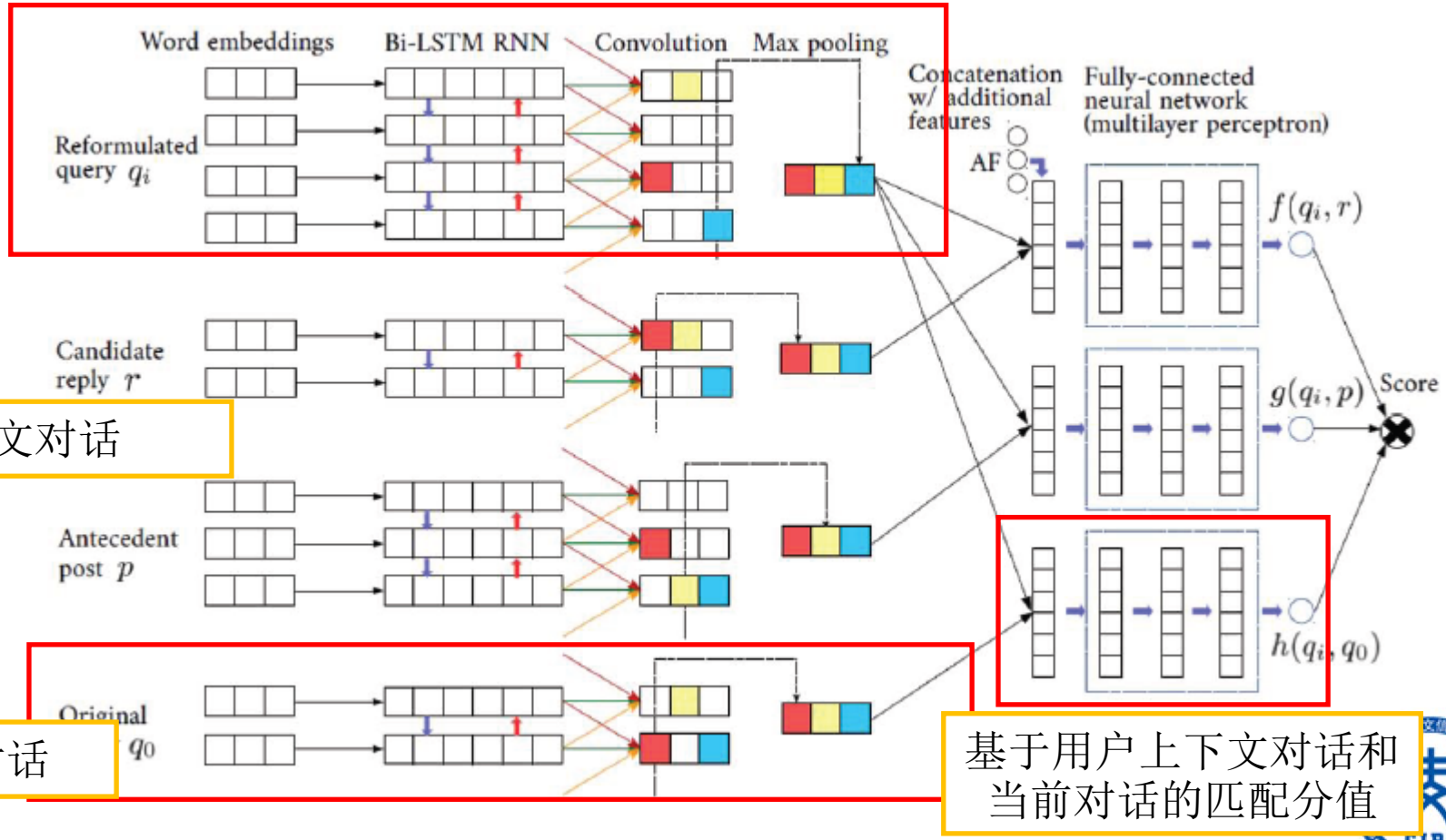


基于用户上下文对话和候选回答的提问的匹配分值

用户文本理解

- 基于深度学习的人机对话系统 [Yan et al., SIGIR 2016]
- 通过借助对话上下文信息帮助理解**用户对话意图**

Posting: 近视了需要戴眼镜...
(I need a pair of glasses because of the myopia...)
Reply$_1$: 我送你眼镜！
(I will offer the glasses for you!)
Reply$_2$: 可以恢复的，别紧张...
(You will be recovered. Don't worry.)



用户上下文对话

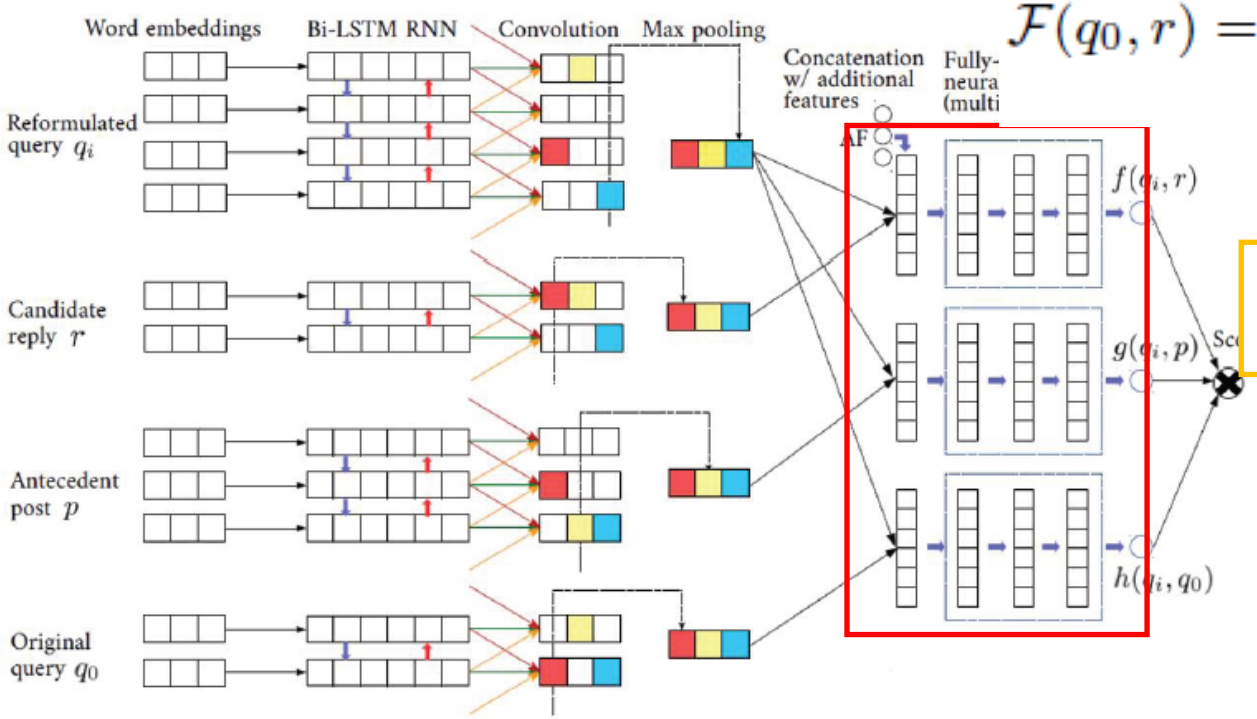
~~Human-Computer Conve~~
A_1: 天哪一把年纪的人居然近视了
(OMG I got myopia at such an "old" age)
B_1: 真的吗？
(Really?)
A_2:嗯哪。求个眼镜做礼物！
(Yeah. Wish a pair of glasses as ...if.)
B_2:我送你眼镜！
(I will offer the glasses for you!)

用户当前对话

基于用户上下文对话和当前对话的匹配分值

用户文本理解

- 基于深度学习的人机对话系统 [Yan et al., SIGIR 2016]
- 通过借助对话上下文信息帮助理解**用户对话意图**



$$\mathcal{F}(q_0, r) = \sum_{i=0}^{|Q|} \left(h(q_0, q_i) \sum_{p} \left(f(q_i, r) \cdot g(q_i, p) \right) \right)$$

基于用户上下文对话和当前对话的匹配分值

基于用户上下文对话和候选回答的提问的匹配分值

基于用户上下文对话和候选回答的匹配分值

用户文本理解

- 基于深度学习的人机对话系统 [Yan et al., SIGIR 2016]
- 实验数据：
  - 通过社交媒体抓取对话文本数据<对话,回答>
  - 百度知道、豆瓣论坛、百度贴吧、新浪微博
- 模型训练：
  - 基于负采样的随机梯度下降, minibatch = 100
  - BLSTM使用128维、CNN使用256维

Source	#Posting	#Reply	#Vocabulary
Zhidao	8,915,694	3,705,302	1,499,691
Douban	10,618,981	2,963,226	483,846
Tieba	4,189,160	3,730,248	1,046,130
Weibo	186,963	393,654	119,163
Misc.	3,056	1,548	4,297
Total	9,023,854	7,293,978	2,857,378

$$\underset{\Omega}{\text{minimize}} \sum_{q_0, r^+} \max\{0, \Delta + \mathcal{F}(q_0, r^+) - \mathcal{F}(q_0, r^-)\} + \lambda \|\Omega\|_2^2$$

- 测试数据：
  - 11,097对话
  - 人工判断回答的准确性

用户文本理解

- 基于深度学习的人机对话系统 [Yan et al., SIGIR 2016]
- 性能对比

Model	p@1	MAP	nDCG@5	nDCG@10	nDCG@20	MRR
SMT (Ritter et al., [26])	0.363					
LSTM-RNN (Sutskever et al., [32])	0.441					
NRM (Shang et al., [29])	0.465					
Random Match	0.266	0.246	0.247	0.289	0.353	0.083
Okapi BM25	0.272	0.253	0.337	0.302	0.368	0.169
DeepMatch (Lu and Li, [17])	0.457	0.317	0.419	0.454	0.508	0.275
LSTM-RNN (Palangi et al., [25])	0.338	0.283	0.330	0.371	0.431	0.228
ARC (Hu et al., [7])	0.394	0.294	0.397	0.421	0.477	0.232
DeepMatch w/ context adaption	0.603	0.378	0.555	0.584	0.628	**0.349**
LSTM-RNN w/ context adaption	0.362	0.296	0.354	0.395	0.453	0.237
ARC w/ context adaption	0.400	0.309	0.383	0.422	0.480	0.319
Deep Learning-to-Respond (DL2R)	**0.731***	**0.416***	**0.663***	**0.682***	**0.717***	0.333

与其他基于深度学习的模型相比，基于BOW的BM25性能最差

用户文本理解

- 基于深度学习的人机对话系统 [Yan et al., SIGIR 2016]
- 性能对比

Model	p@1	MAP	nDCG@5	nDCG@10	nDCG@20	MRR
SMT (Ritter et al., [26])	0.363					
LSTM-RNN (Sutskever et al., [32])	0.441					
NRM (Shang et al., [29])	0.465					
Random Match	0.266	0.246	0.247	0.289	0.353	0.083
Okapi BM25	0.272	0.253	0.337	0.302	0.368	0.169
DeepMatch (Lu and Li, [17])	0.457	0.317	0.419	0.454	0.508	0.275
LSTM-RNN (Palangi et al., [25])	0.338	0.283	0.330	0.371	0.431	0.228
ARC (Hu et al., [7])	0.394	0.294	0.397	0.421	0.477	0.232
DeepMatch w/ context adaption	0.603	0.378	0.555	0.584	0.628	**0.349**
LSTM-RNN w/ context adaption	0.362	0.296	0.354	0.395	0.453	0.237
ARC w/ context adaption	0.400	0.309	0.383	0.422	0.480	0.319
Deep Learning-to-Respond (DL2R)	**0.731***	**0.416***	**0.663***	**0.682***	**0.717***	0.333

基于BLSTM+CNN的模型效果最佳

用户文本理解

- 基于深度学习的人机对话系统 [Yan et al., SIGIR 2016]
- 性能对比

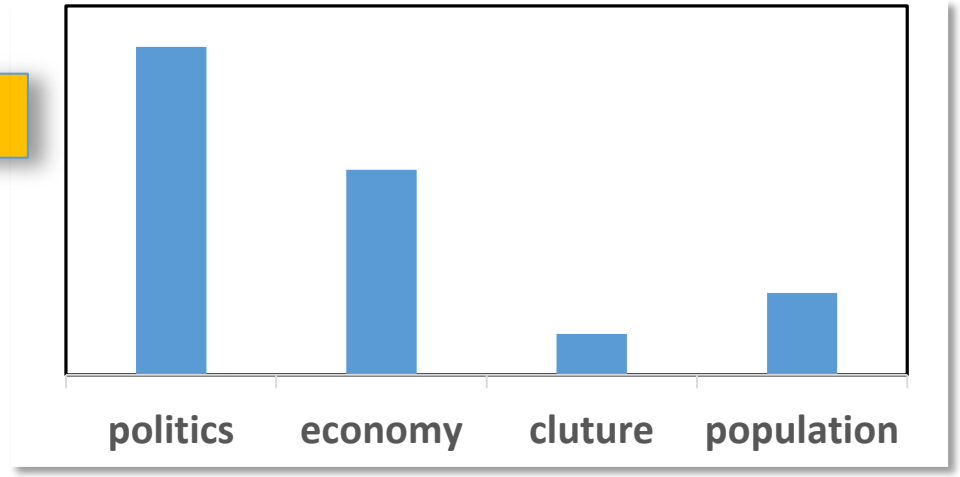
	p@1	MAP	nDCG@5	nDCG@10	nDCG@20	MRR
Query-Reply w/o Query-Context	0.522	0.340	0.476	0.509	0.559	0.296
Query-Posting w/o Query-Context	0.510	0.302	0.404	0.425	0.489	0.285
Query-Reply w/ Query-Context	0.596	0.366	0.528	0.561	0.603	0.327
Query-Posting w/ Query-Context	0.563	0.362	0.483	0.516	0.568	0.316
Full Combination	0.731	0.416	0.663	0.682	0.717	0.333

结合用户**上下文对话记录**和**候选回答的提问信息**能提供更过的语义信息，
有效地帮助确定正确的回答

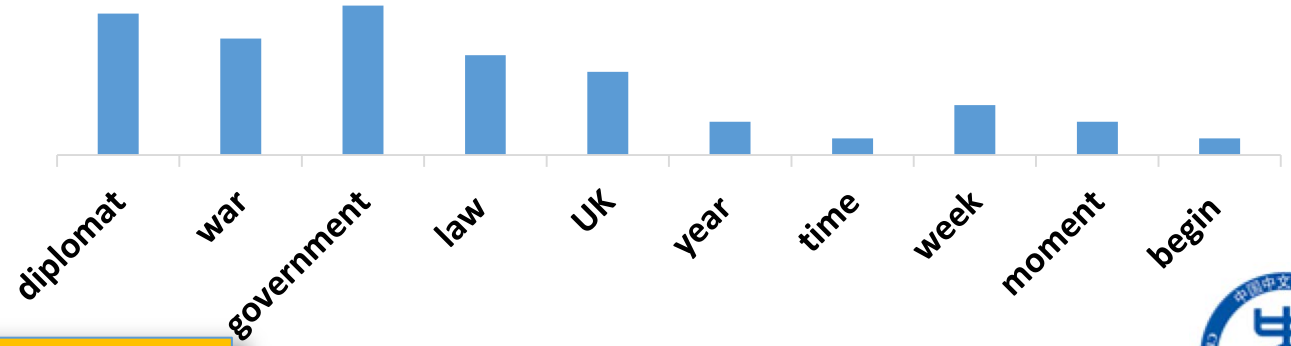
用户文本理解

- 基于深度学习的文本主题模型 [Cao et al., AAAI 2015]
- 文本主题模型

Each document has a topic distribution



Politics



Each topic has a word distribution

用户文本理解

- 基于深度学习的文本主题模型 [Cao et al., AAAI 2015]
- 文本主题模型
  - 单词基于文本的条件概率

$$p(w|d) = \sum_{i=1}^{K} p(w|t_i)p(t_i|d),$$

$$p(w|d) = \phi(w) \times \theta^T(d)$$

单词w在各个主题下的分布概率

文档d在各个主题下的分布概率

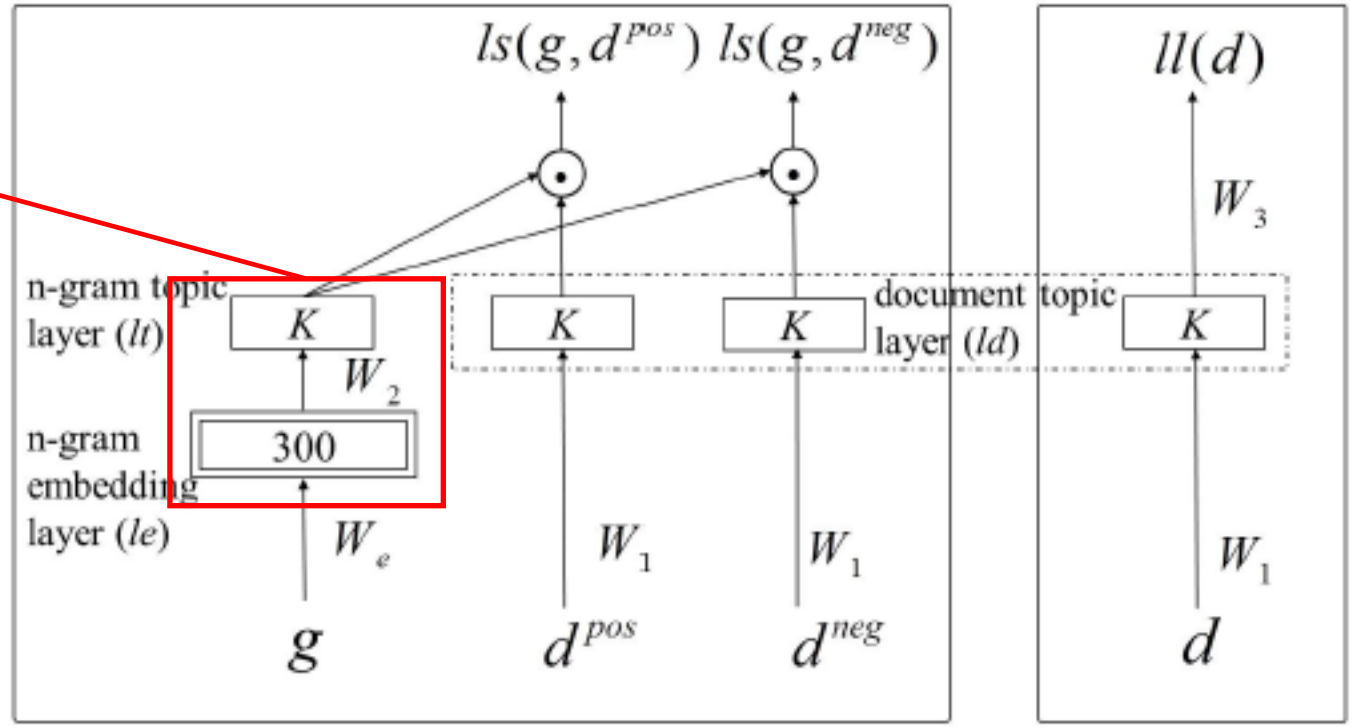
用户文本理解

- 基于深度学习的文本主题模型 [Cao et al., AAAI 2015]
- 通过单词/词组的向量表达，计算单词的主题分布概率

$$lt(g) = sigmoid(le(g) \times W_2)$$

单词/词组g的向量由Word2Vec学习得到

单词w在各个主题下的分布概率



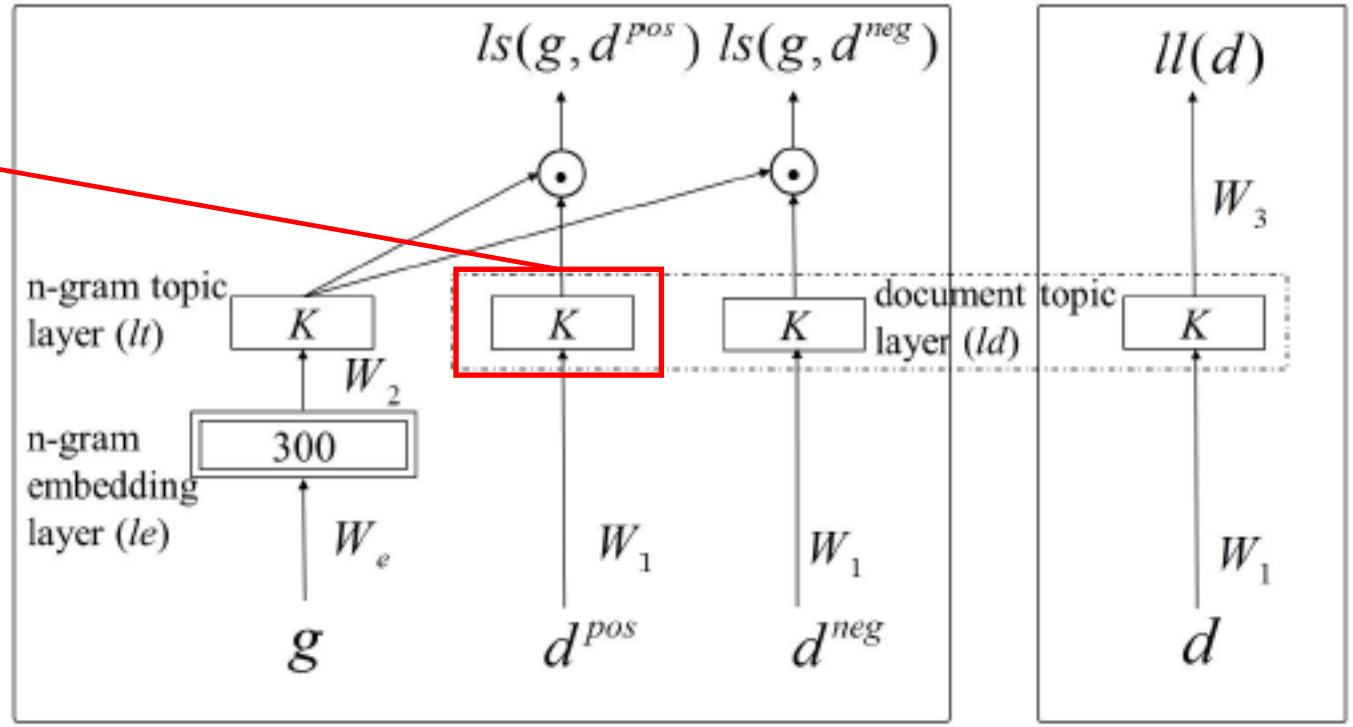
用户文本理解

- 基于深度学习的文本主题模型 [Cao et al., AAAI 2015]
- 通过单词/词组的向量表达，计算单词的主题分布概率

$$ld(d) = softmax(W_1(d,:))$$

文档d的主题分布

文档d的向量表达



用户文本理解

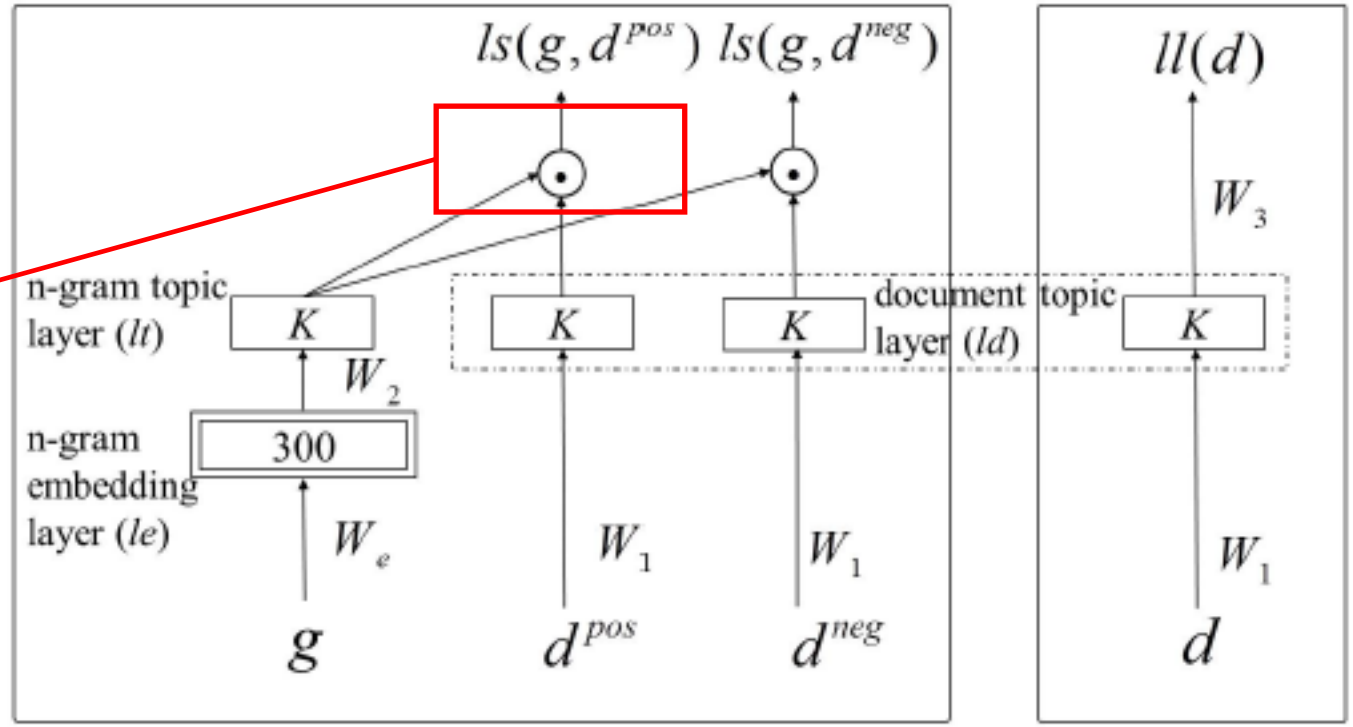
- 基于深度学习的文本主题模型 [Cao et al., AAAI 2015]
- 通过单词/词组的向量表达，计算单词的主题分布概率

$$ld(d) = softmax(W_1(d,:))$$

$$lt(g) = sigmoid(le(g) \times W_2)$$

$$ls(g,d) = lt(g) \times ld(d)^T$$

文档d与单词/词组g的的匹配分值

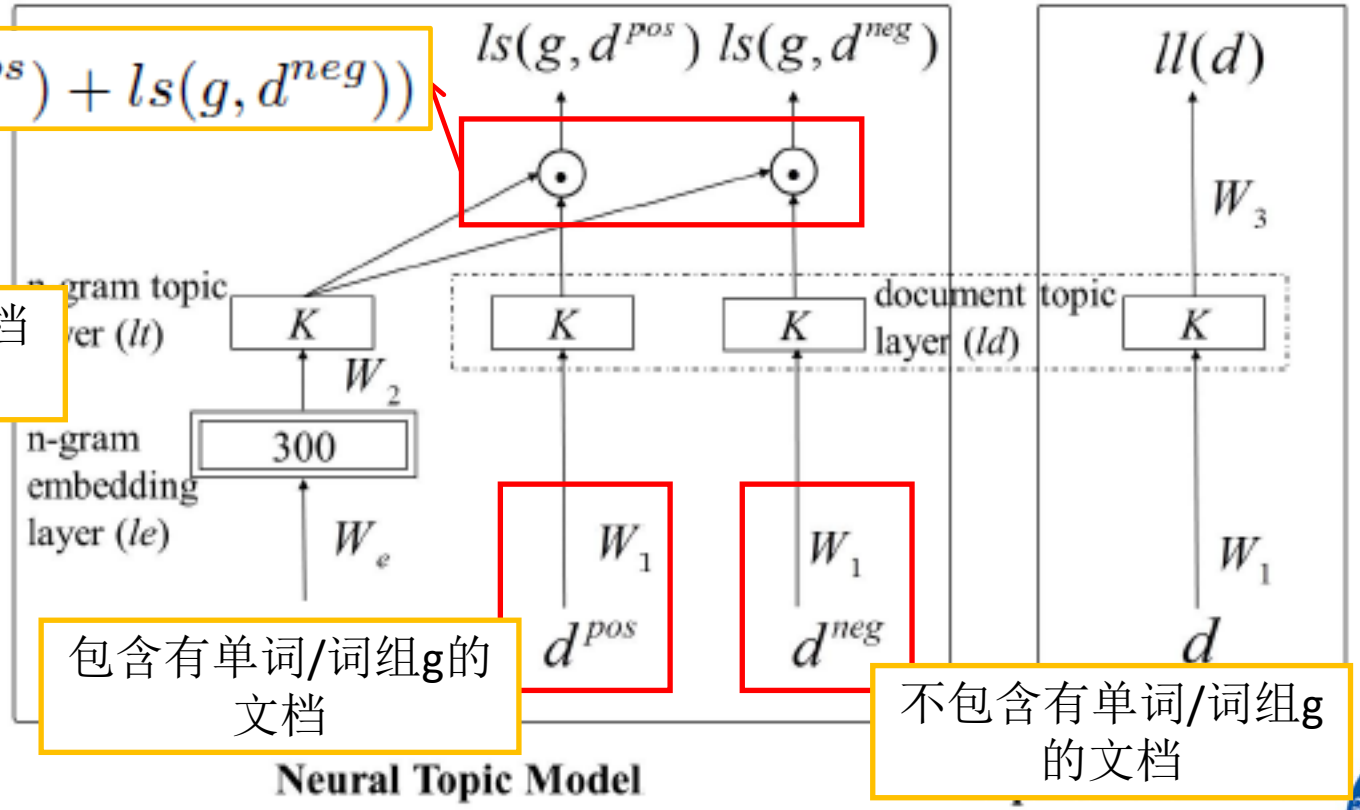


用户文本理解

- 基于深度学习的文本主题模型 [Cao et al., AAAI 2015]
- 通过单词/词组的向量表达，计算单词的主题分布概率

$$c(g, d^{pos}, d^{neg}) = \max(0, \Omega - ls(g, d^{pos}) + ls(g, d^{neg}))$$

基于负采样的思想，包含有单词/词组g的文档匹配分值应该更高



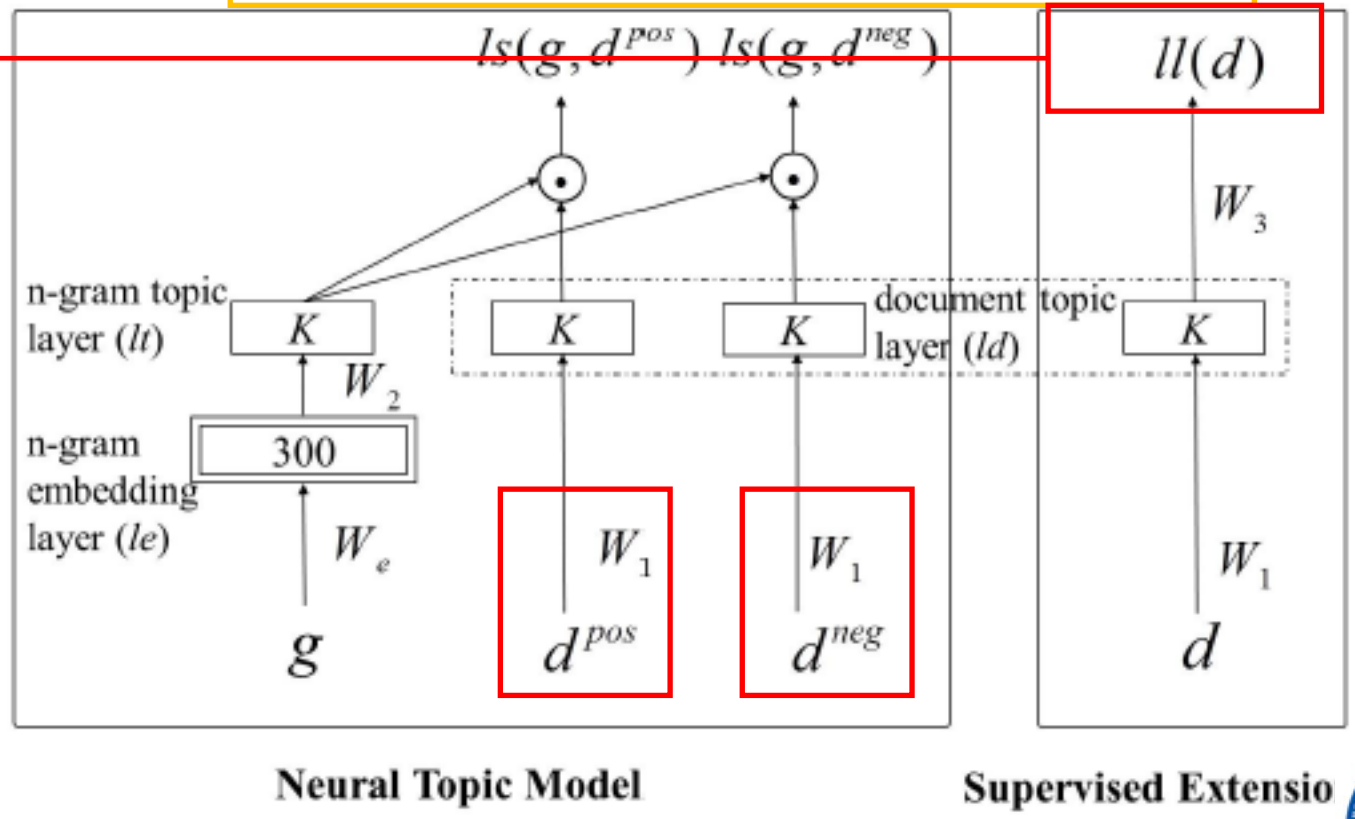
用户文本理解

- 基于深度学习的文本主题模型 [Cao et al., AAAI 2015]
- 通过单词/词组的向量表达，计算单词的主

利用额外标签/类别信息，将模型转变为监督型主题模型

$$ll(d) = f(ld(d) \times W_3)$$

文档的监督信息



用户文本理解

- 基于深度学习的文本主题模型 [Cao et al., AAAI 2015]
- 实验数据：

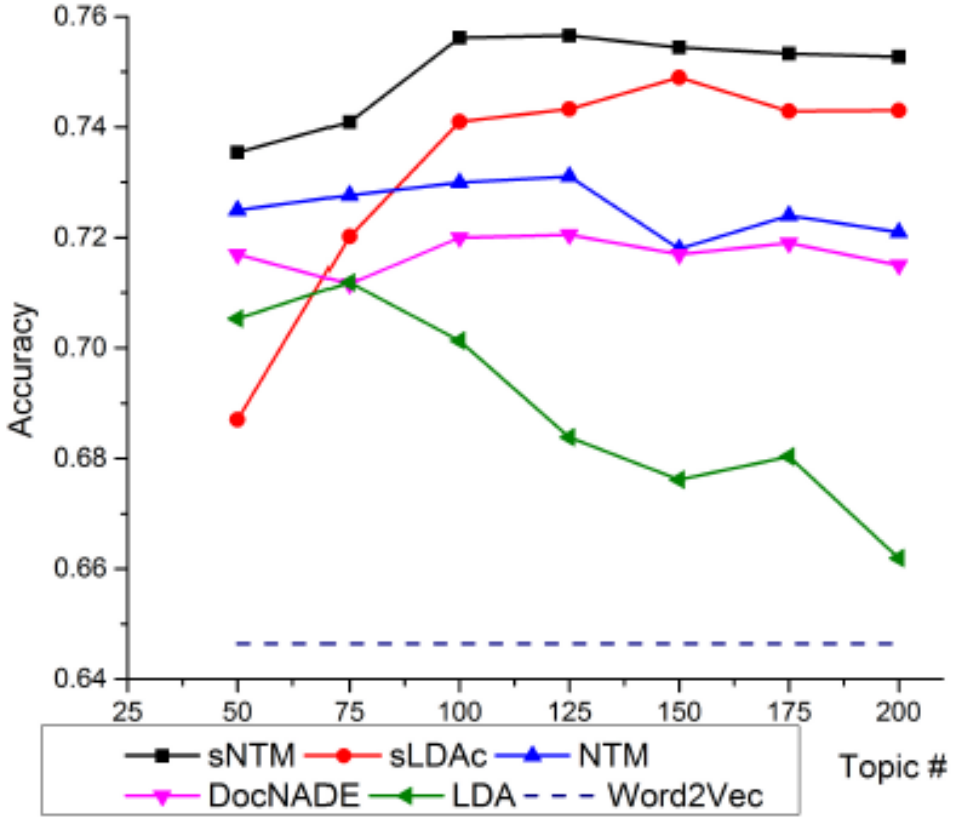
Dataset	20 Newsgroups	Wiki10+	Sentiment Scale
Train.Size	11149	11550	3337
Test.Size	7403	5775	1669
Avg. Length	135	1704	176
Task	MCC	MLC	Regression

- 对比方法：
  - sNTM：基于深度学习的监督型主题模型 （unigram/bigram）
  - NTM：基于深度学习的无监督型主题模型（unigram/bigram）
  - LDA：标准LDA主题模型
  - sLDAc：基于LDA的监督型主题模型
  - Word2Vec：基于词向量表达的模型

用户文本理解

- 基于深度学习的文本主题模型 [Cao et al., AAAI 2015]
- 对比方法：
  - sNTM：基于深度学习的监督型主题模型
  - NTM：基于深度学习的无监督型主题模型
  - LDA：标准LDA主题模型
  - sLDAc：基于LDA的监督型主题模型
  - Word2Vec：基于词向量表达的模型

基于深度学习的监督型主题模型sNTM取得最好的分类效果；
基于深度学习的无监督型主题模型明显优于其他无监督型模型



用户文本理解

- 基于深度学习的文本主题模型 [Cao et al., AAAI 2015]
- 生成主题对比

Class 1(alt.atheism)			Class 16(soc.religion.christian)			Class 20(talk.religion.misc)		
LDA(4)	DocNADE(31)	NTM(47)	LDA(66)	DocNADE(27)	NTM(59)	LDA(5)	DocNADE(8)	NTM(72)
belief	moral	atheism	jesus	jesus	god	lds	church	lds church
god	atheism	creation	god	god	jesus	istanbul	catholic	pope
truth	keith	human being	christ	sin	catholic faith	georgia	orthodox	protestant
reason	religion	true nature	s	lord	christ	ermeni	holy	nestorius
atheist	system	science	lord	heaven	saint	york	tradition	religious belief
christian	islam	sacred scripture	sin	church	bible	church	doctrine	catholic church
bible	belief	theism	bible	life	homosexuality	ankara	movement	religious fanaticism
religion	muslim	proof	heaven	hell	heaven	##	spirit	theology

LDA产生的主题词并不全部有意义，不少单词仅仅是数字或者无语义信息。
NTM由于可以考虑bigram，因此其生成的主题词更能放映主题的语义内含。

总结

- 传统文本表示方式
  - One-Hot表达：稀疏、维度大、语义关系缺失
  - BoW词袋模型：丢失单词间的组合次序
- 深度学习
  - 稠密的实数向量表达：包含语义/语法关系，维度固定。
    - 可以很好的对词组进行特征表达
  - RNN模型
    - **保留句子中单词的组合次序**
    - 对序列进行特征抽取，可处理**变长的序列数据**
  - CNN模型
    - 可处理变长的数据，适用于**词语组合**
    - 基于局部特征构造全局特征，池化过程类似关注机制
- 关注机制
  - 类似TF-IDF，**区分重要信息与非重要信息**

Thanks

Question&Answer!