



AI CHALLENGER  
全球AI挑战赛 2018

# 细粒度用户评论情感分析 参赛队：simtony

欧泽彬 追一科技

# 目录



- 问题描述
- 模型架构演变
- 模型介绍和消融分析
  - 模型结构
  - 训练过程的 Babysitting
- 结论

# 目录



- 问题描述
- 模型架构演变
- 模型介绍和消融分析
  - 模型结构
  - 训练过程的 Babysitting
- 结论

# 任务和数据统计特性



- 给定用户评论，推断其20个aspect的情感标签
    - 5个粗粒度大类
    - 20个细粒度小类
  - Train/Valid 分布：标签占比严重不均衡



# 数据和问题特性

- 标注噪声大
- 存在无关信息
- 可能需要常识

大众点评就是给力呀，又让我中了一次霸王餐～可让我兴奋了！！虽然这次中的离我比较远，我在南昌市里，店在莲塘，今天，天公又不作美，下了一天的雨，以前也中过一次霸王餐，也合适我口味，所以我今天早早出发了，第一个到店里的，再怎么远都必须得去！！这家店位于沃尔玛的右侧的巷子里，停车也方便～环境属于简单大方型的，环境非常ok～很干净，服务态度也很棒～老板特别大方，我们十二个人，个个都吃的非常开心，吃的非常饱～太赞了～来说说今天让我印象比较深的几个菜，就简单说说吧～第一个菜名是“五味飘香煲”里面有水晶包，鸭掌，鸭舌，豆腐干！还有一宝我不记得了，原谅我记性差哈～水晶包不油腻，鸭掌和鸭舌都是卤过的，味道很不赖！鸭掌还去掉了骨头，吃起来非常方便～第二个菜名是“板栗烧鸡”里面的板栗很粉很好吃，鸡处理的也很不错。腥味没有，肉质也不老，刚刚好～也不错的一道菜～值得尝试！第三个菜名是“小炒鹅脯肉”这鹅肉是烟熏过的～鹅肉的肉质比鸡肉，鸡肉好多了，没有松松垮垮的，比较紧！味道我喜欢吃～第四个是叫什么牛蛙的，牛蛙处理的很好，肉很嫩，也不错，反正他家菜式做的很不错，大家可以去尝试一下哟，我在这里给个意见，就是那个沙拉香蕉鱼卷，总体味道很不错，就是如果去掉里面的鱼刺就更加完美了～～～加油，这家店非常攒～～

层次一	层次二	情感标签
位置	交通是否便利	负向
	距离商圈远近	未提及
	是否容易寻找	负向
服务	排队等候时间	未提及
	服务人员态度	正向
	是否容易停车	正向
价格	点菜/上菜速度	未提及
	价格水平	正向
	性价比	正向
环境	折扣力度	正向
	装修情况	正向
	嘈杂情况	正向
	就餐空间	正向
菜品	卫生情况	正向
	分量	正向
	口感	正向
	外观	正向
其他	推荐程度	正向
	本次消费感受	正向
其他	再次消费的意愿	正向

# 问题分析

- 问题定义 – 20个任务的多任务问题：
  - 4分类问题
  - 2分类问题 + 3分类问题
- 任务间有相互联系: the physical law
- 输入信息的稀疏：
  - 未对 aspect 的描述进行标注
  - 端到端模型依赖 attention 做 alignment
- 监督信号的稀疏: label 和词的共现
  - 构造更强的训练信号
  - 输入更强的语义特征
  - 防止过拟合

# 目录



- 问题描述
- 模型架构演变
- 模型介绍和消融分析
  - 模型结构
  - 训练过程的 Babysitting
- 结论

# NLP 任务的抽象



词特征:

词向量: Skip-gram, Glove,  
POS embedding

上下文相关: ELMo

Sub-word: character

...

计算词表征:

上下文信息: RNN, CNN,  
Transformer  
Context 信息: Co-Attention,  
Attention

上下文无关: MLP, Highway,  
Residual

...

词表征聚合:

Attention, Pooling  
RNN 最后的 step  
...

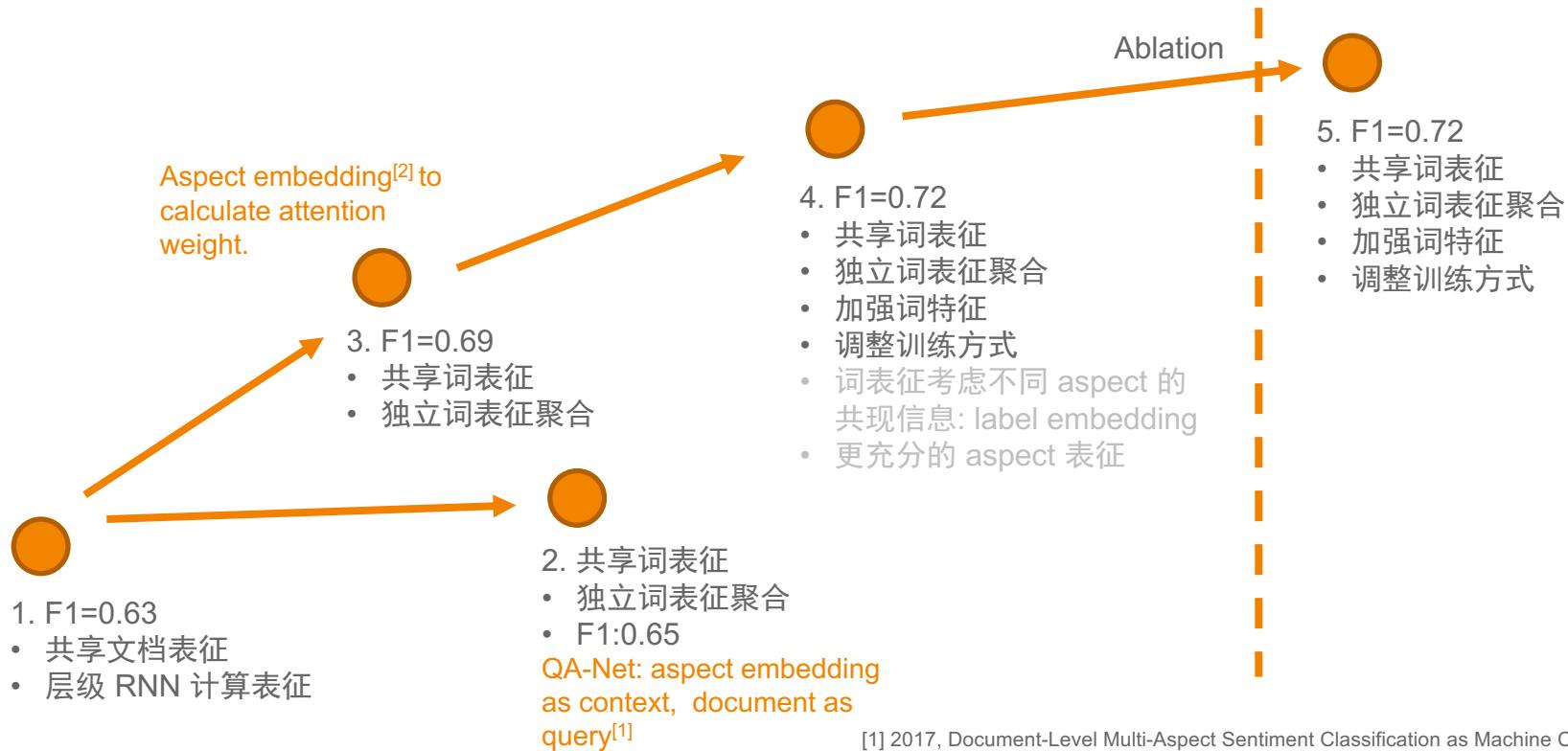
文档表征:

MLP, Highway, Maxout  
...

预测层:

分类  
阅读理解  
生成  
...

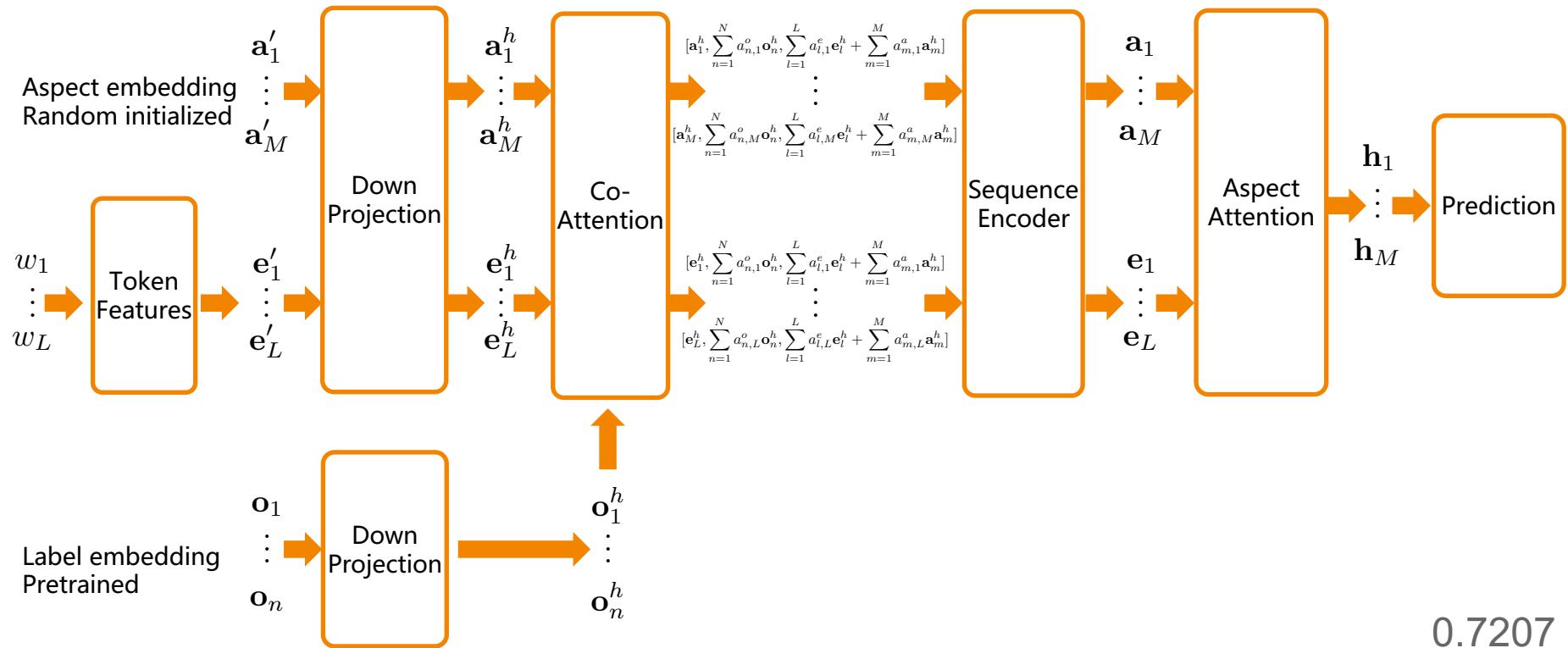
# The Graduate Student Descent



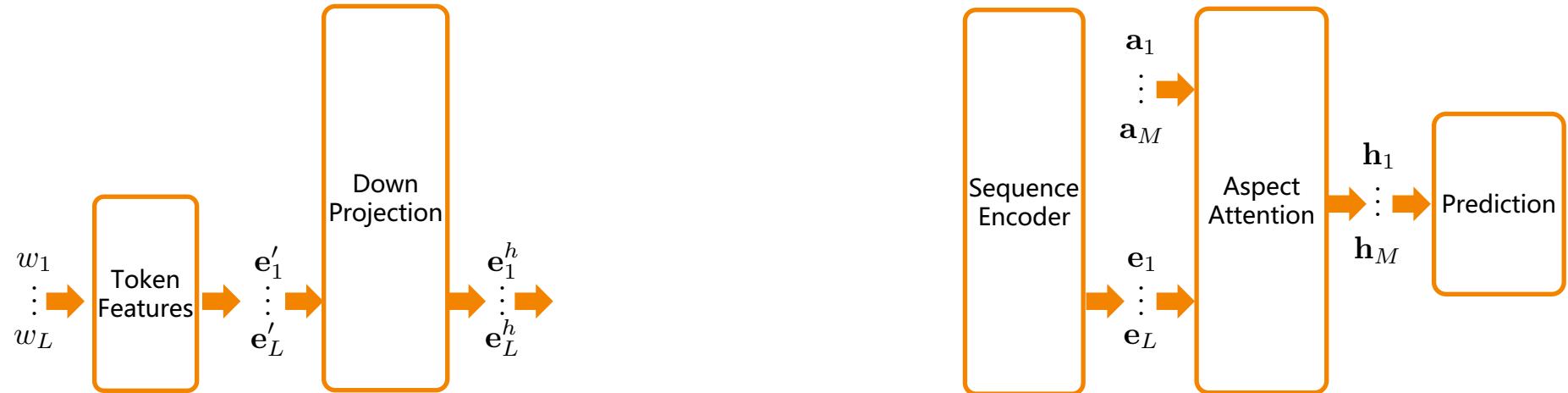
[1] 2017, Document-Level Multi-Aspect Sentiment Classification as Machine Comprehension

[2] 2016, Attention-based LSTM for Aspect-level Sentiment Classification

# 原始模型



# 精简后的模型



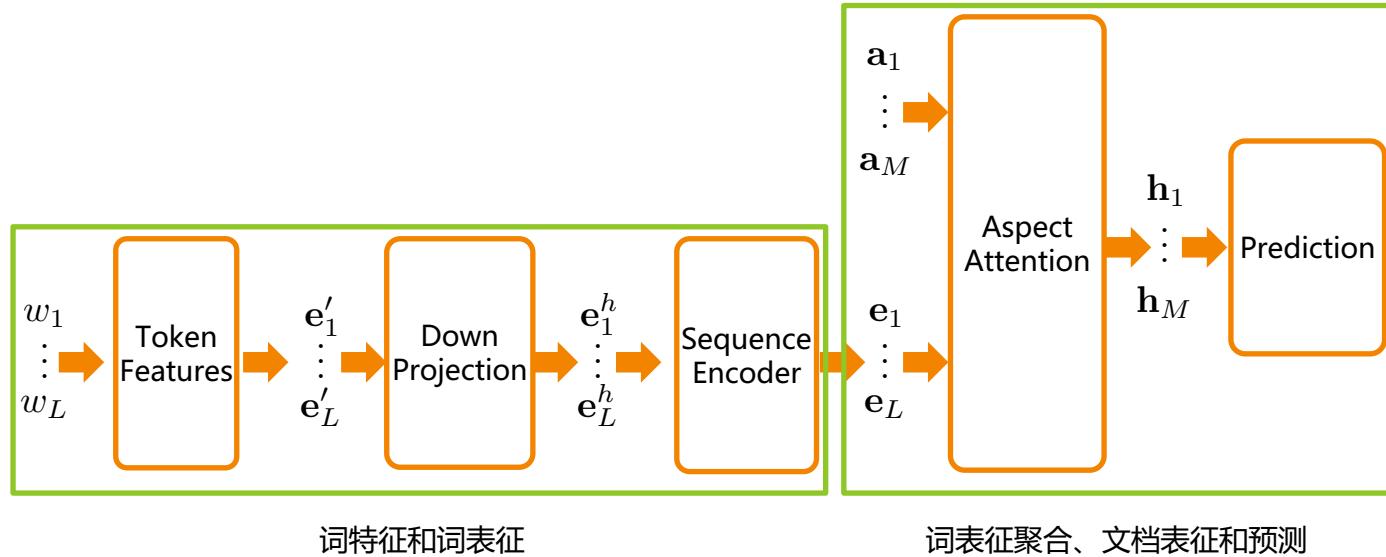
0.7207 -> 0.7220

# 目录



- 问题描述
- 模型架构演变
- 模型介绍和消融分析
  - 模型结构
  - 训练过程的 Babysitting
- 结论

# 精简后的模型

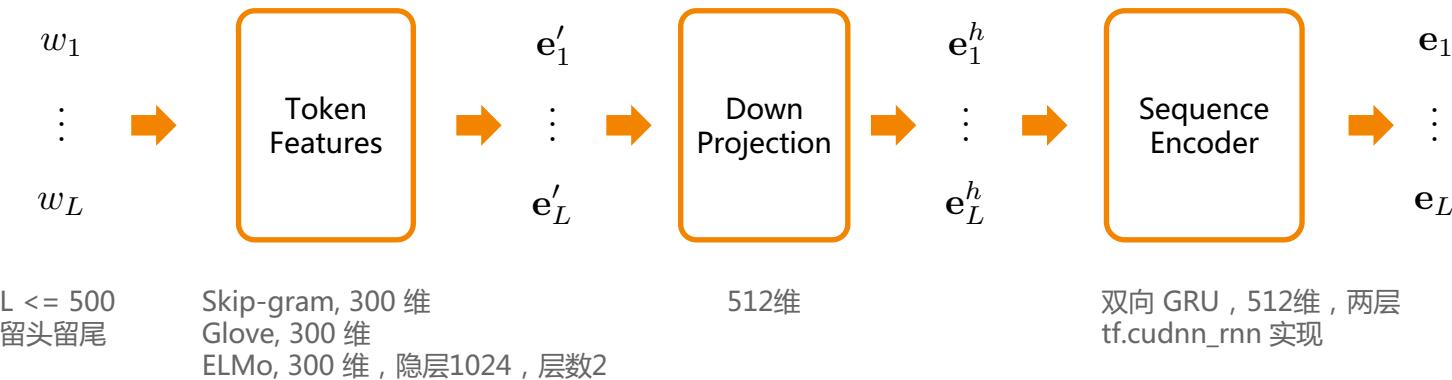


# 目录

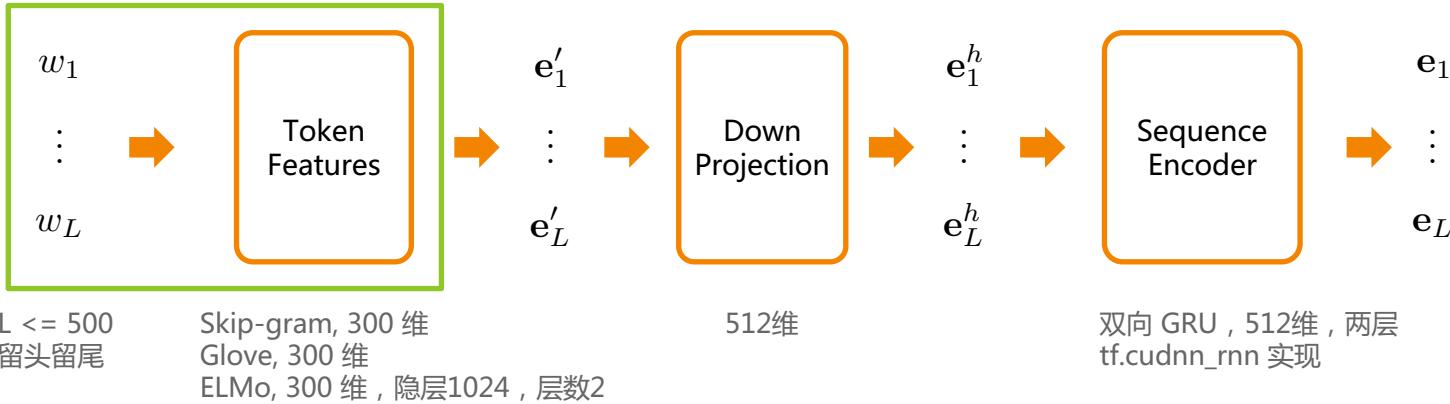


- 问题描述
- 模型架构演变
- 模型介绍和消融分析
  - 模型结构
  - 训练过程的 Babysitting
- 结论

# 模型结构：词特征和词表征



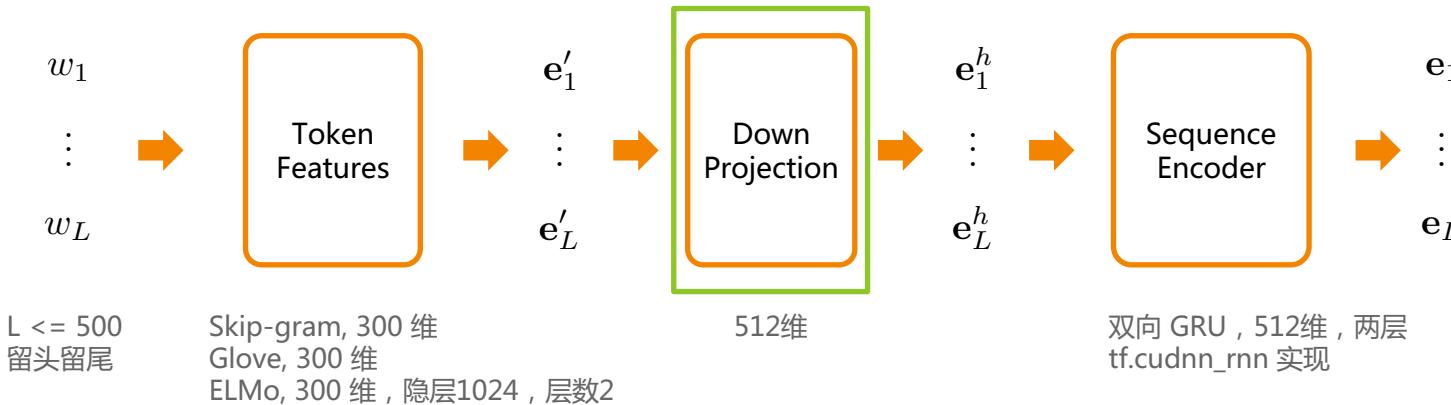
# Token Features



特征	F1 score
Skip-gram + Glove + ELMo	0.7220
只用 ELMo(训练更充分)	0.7208
只用 ELMo	0.7201
只用 Glove	0.7135
只用 Skip-gram	0.7122

2013, Distributed Representations of Words and Phrases and their Compositionality  
 2014, Glove: Global vectors for word representation  
 2018, Deep contextualized word representations

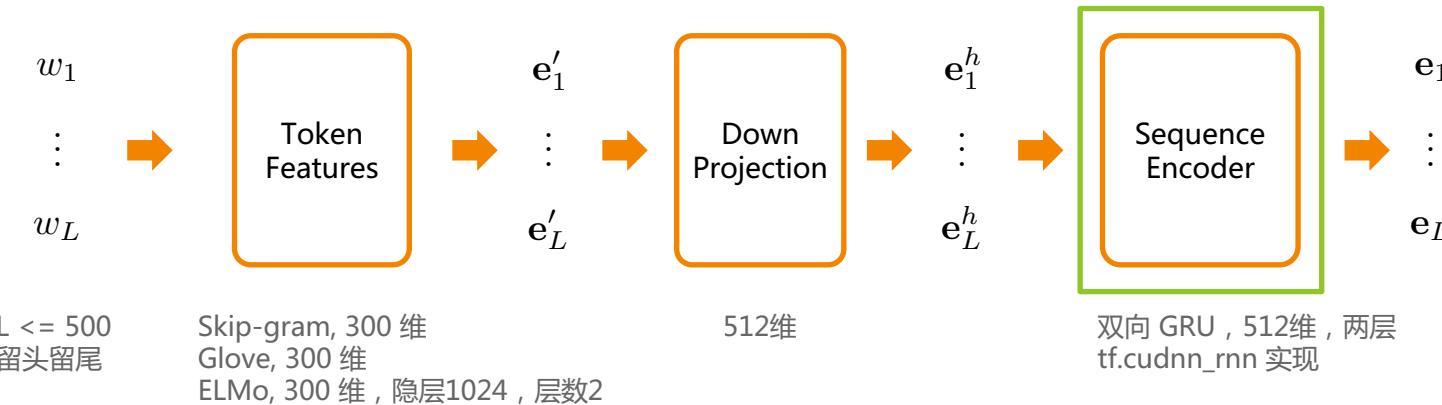
# Down Projection



方案	F1 score
不做再编码	0.7219
两层 highway block	0.7220
两层 MLP	0.7138

Transform:  $\mathbf{h} = \tanh(\mathbf{W}_h \mathbf{x} + \mathbf{b}_h)$   
 Carry Gate:  $\mathbf{c} = \text{sigmoid}(\mathbf{W}_c \mathbf{x} + \mathbf{b}_c)$   
 Output:  $\mathbf{o} = \mathbf{h} \circ (\mathbf{1} - \mathbf{c}) + \mathbf{c} \circ \mathbf{x}$

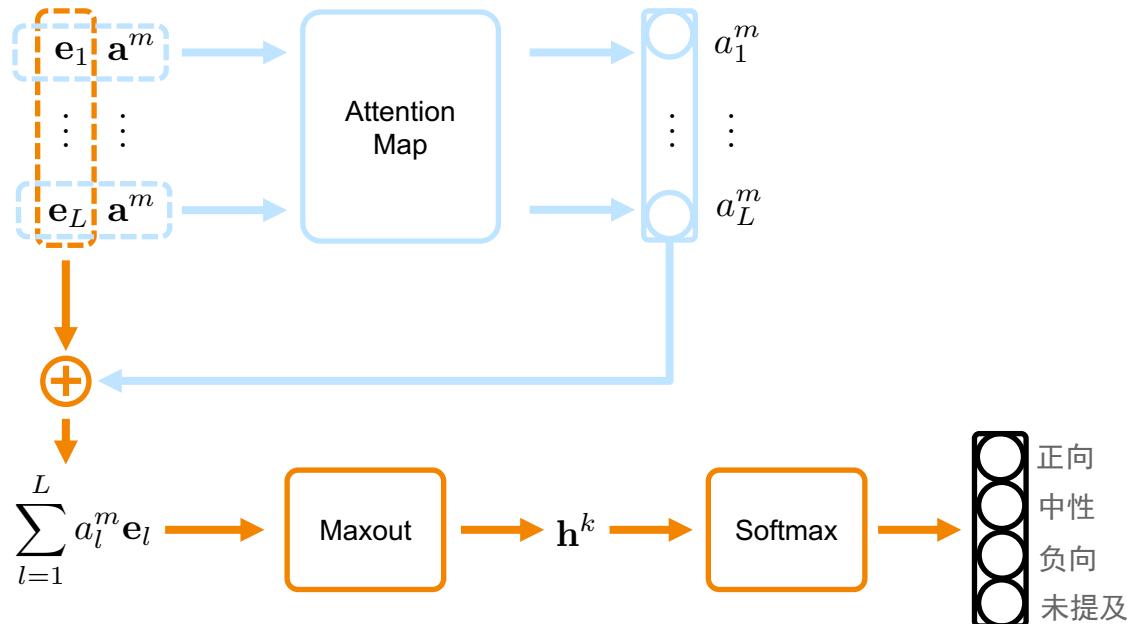
# Sequence Encoder



RNN 层数	F1 score
2	0.7220
1	0.7209

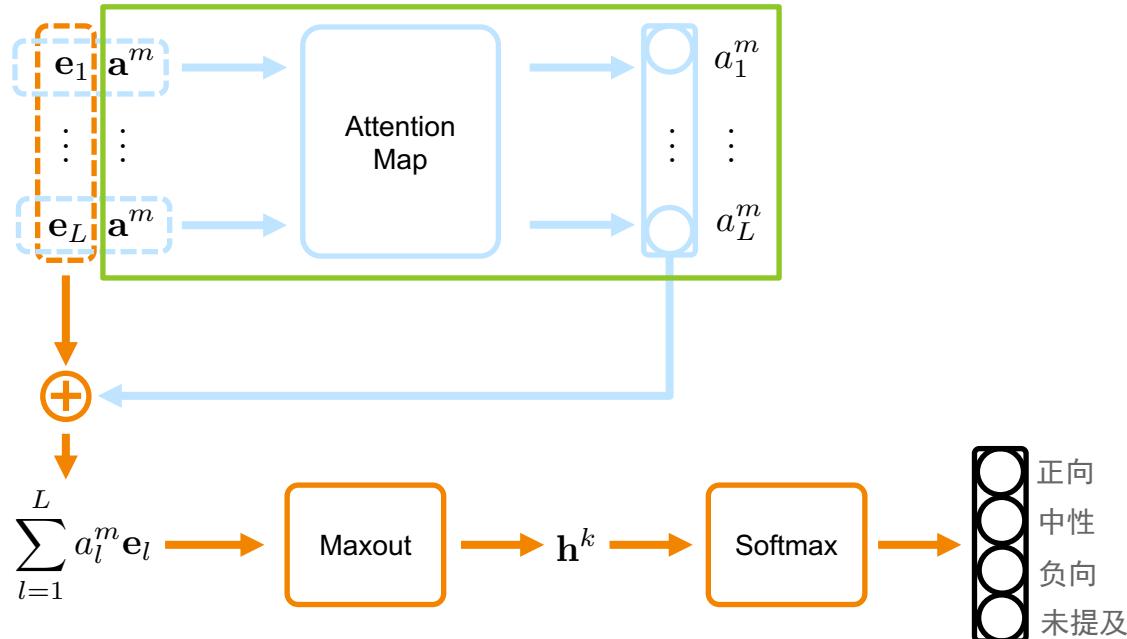
# 模型结构：词表征聚合、文档表征和预测

对于每个 aspect embedding  $a^m$ :



# 词表征聚合 : Attention

对于每个 aspect embedding  $a^m$ :



# 词表征聚合 : Attention



- Attention 的元素
  - Query vector:  $\mathbf{q}$
  - Key vector:  $\mathbf{k}_l = \sigma(\mathbf{e}_l)$
  - Query-key Similarity:  $s_l = \mathbf{q}^T \mathbf{k}_l$
  - Normalized Weights:  $a_l = \frac{\exp(s_l)}{\sum_{l=1}^L \exp(s_l)}$

# 词表征聚合 : Attention

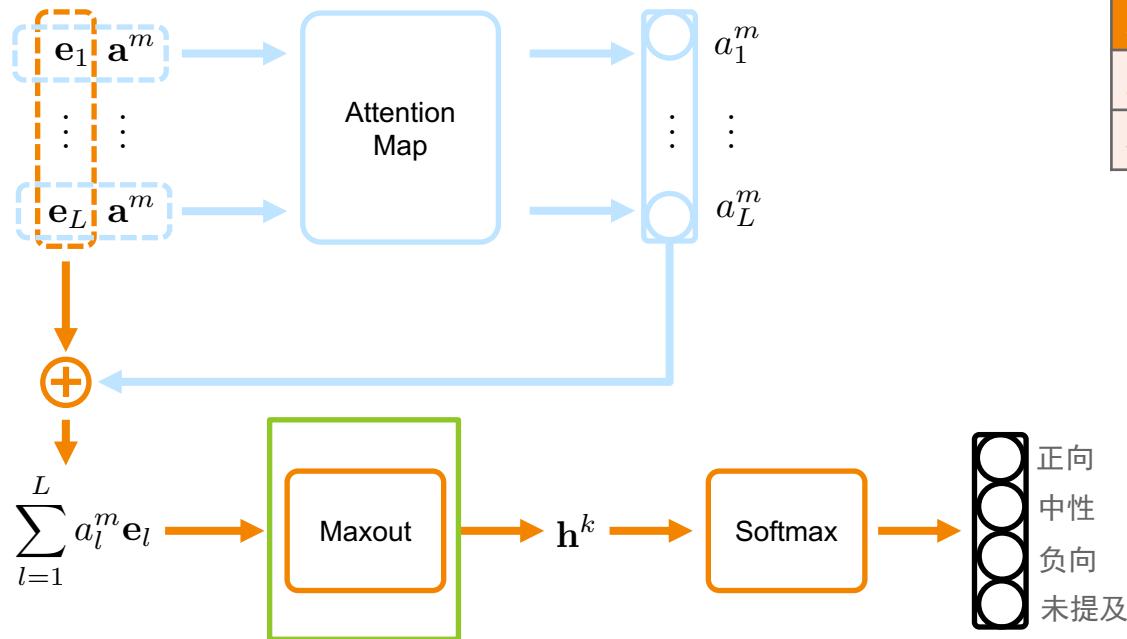


- 怎么计算 key vector
  - Baseline:  $k_l = \tanh(\mathbf{W}\mathbf{e}_l + \mathbf{b})$
  - 用 Aspect embedding 加入 aspect 间独立性:  $k_l = \tanh(\mathbf{W}[\mathbf{e}_l, \mathbf{a}_m] + \mathbf{b})$
- Attention 参数在 aspect 间共享的方式: q, W, b

W, b, q 共享方案	有 aspect embedding F1 score	无 aspect embedding F1 score
所有类共享	0.7223	0.7188
粗粒度大类共享	0.7220	0.7206
不共享	0.7213	0.7213

# 文档表征和预测

对于每个 aspect embedding  $a^m$ :



方案	F1 score
加 Maxout	0.7220
不加 Maxout	0.7205

# 目录

- 问题描述
- 模型架构演变
- 模型介绍和消融分析
  - 模型结构
  - 训练过程的 Babysitting
- 结论

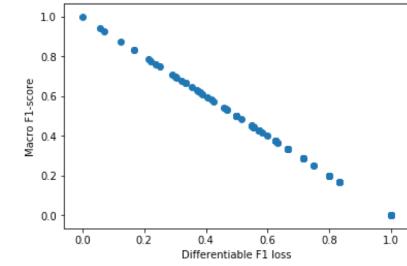
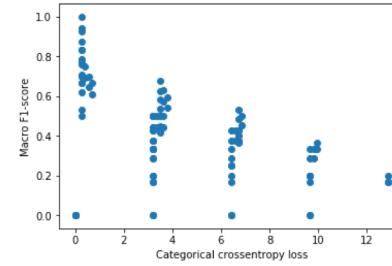
# 训练过程的 Babysitting

- 辅助任务: F1 loss
  - Cross Entropy 和 F1 score 并不一一对应
- 对参数的 Exponential moving average(EMA)：
  - 平滑因梯度估计噪声造成的参数抖动
- Cyclic learning rate scheduling
  - Warm restart 防止过早进入局部最优

# 辅助任务: F1 loss

- Better proxy for F1 metric:

备注	公式	F1 score
Focal loss style	$1-(1-F1)*F1$	0.7220
Basic F1 loss	$1-F1$	0.7209
No F1 loss	N/A	0.7186
Log F1 loss		
Focal loss style Log F1 loss		



Predict vector:  $\hat{\mathbf{y}} = [p_1, p_2, p_3, p_4], p_k \in [0, 1], \sum_k p_k = 1$

Label vector:  $\mathbf{y} = [i_1, i_2, i_3, i_4], i_k \in \{0, 1\}, \sum_k i_k = 1$

Presision:  $\mathbf{p} = \frac{\sum_k p_k i_k}{\sum_k p_k i_k + \sum_k p_k (1 - i_k)}$

Recall:  $\mathbf{r} = \frac{\sum_k p_k i_k}{\sum_k p_k i_k + \sum_k (1 - p_k) i_k}$

F1:  $\mathbf{f1} = \text{harmonic\_mean}(\mathbf{p}, \mathbf{r})$

# EMA: Exponential moving average

- 减小梯度更新噪声，对于每步更新：

$$\hat{v}_t = \alpha \hat{v}_{t-1} + (1 - \alpha)v_t$$

- 参数的影响：

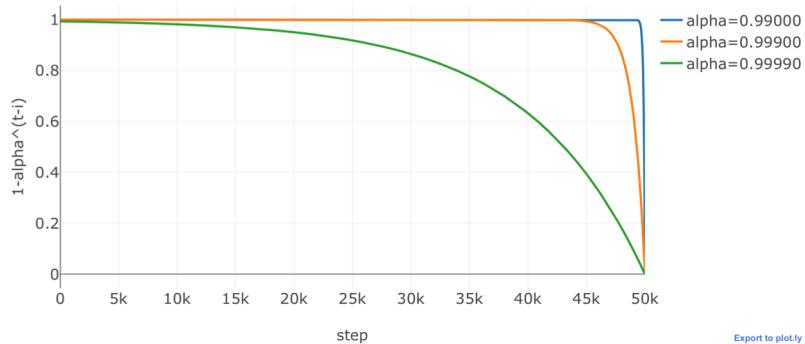
Alpha 取值	F1 score
0(No EMA)	
0.99	
0.999	
0.9999	

# EMA 的分析

- 等价于 offline 的 learning rate annealing: avoid sharp minima
- 在第  $t$  个 step: Vanilla:  $v_t = v_1 - \sum_{i=1}^{t-1} g_i$

$$\text{EMA: } \hat{v}_t = (1 - \alpha^t)v_1 - \sum_{i=1}^{t-1}(1 - \alpha^{t-i})g_i$$

Gradient weights



# EMA 的推导

$$\begin{aligned}\hat{v}_t &= \alpha \hat{v}_{t-1} + (1 - \alpha) v_t \\&= \alpha(\alpha \hat{v}_{t-2} + (1 - \alpha) v_{t-1}) + (1 - \alpha) v_t \\&= \dots \\&= (1 - \alpha)(v_t + \alpha v_{t-1} + \alpha^2 v_{t-2} + \dots + \alpha^{t-1} v_1) \\&= (1 - \alpha)(v_1 - \sum_{i=1}^{t-1} g_i + \alpha(v_1 - \sum_{i=1}^{t-2} g_i) + \dots + \alpha^{t-2}(v_1 - \sum_{i=1}^1 g_i) + \alpha^{t-1} v_1) \\&= (1 - \alpha)\left(\frac{1 - \alpha^t}{1 - \alpha} v_1 - \sum_{i=1}^{t-1} \frac{1 - \alpha^{t-i}}{1 - \alpha} g + i\right) \\&= (1 - \alpha^t) v_1 - \sum_{i=1}^{t-1} (1 - \alpha^{t-i}) g_i\end{aligned}$$

# Ensemble



- 对输出的概率取平均
- 多样性策略：
  - 变化 Sequence encoder: RNN, TCN, Transformer
  - 训练集切成多分训练模型
  - 每个 aspect 各取一个最高的 checkpoint
- Ensemble 效果(提交的模型):
  - 单模型最好: 0.7207
  - Ensemble: 0.7247

# 目录

- 问题描述
- 模型架构演变
- 模型介绍和消融分析
  - 模型结构
  - 训练过程的 Babysitting
- 结论

# 结论

- With my best effort:
  - ELMo 效果拔群，但速度很慢: ~0.01
  - F1 loss 很管用: ~0.004
  - EMA 很靠谱: ~
- 精简模型前单模型 valid 0.7207, 精简后单模型 valid 0.7220

#	Team Name	Team Members	Best Score	Entries	Last
1	后厂村静静		0.72946	2	2018/11/15 21:21:30
2	do something	 	0.72794	2	2018/11/15 21:35:36
3	simtony		0.72736	2	2018/11/15 20:43:58

# 心得：工程



- 开始迭代时要预留接口给 ensemble
- 加快迭代速度：
  - 高效的并行化实现
  - 实验队列
- 版本控制和实验管理
- 从最简单的 State of the Art 开始: BiLSTM+Attention

- Tricks are not elegant, but work!
  - 特征增强
- Inductive bias is hard to encode
  - 模型设计和训练的耦合: learnable in theory/learnable in practice
  - Inductive bias 来自于对 domain 的深入理解
- Good empirical research?
  - 排除 Variance 的干扰
  - Ablation analysis
  - 更强的结论：多个参数组合/结构/任务下大概率起作用

# Thanks for your attention!

