Training with streaming annotation

2020年2月14日 0:22

#总结：针对带有标注的数据分批到来并且每个mini-batch数据质量不同的场景，设计了一个在streaming annotated data场景中学习的方法（在NLP领域的事件提取任务上做的实验）。
方法核心：用预训练的transformer网络保存历史batches中重要的信息，并且将历史重要信息和当前batch整合，默认当前batch更重要。

本文特色总结:
1）较为新的研究了按照small-scale batch 以stream形式到达数据的学习策略，可以扩展到除了NLP其他的领域，特别是标注难度大时间长的任务；
2）在历史batch噪声较大的情况下， 本文方法相对于其他方法有明显的优势；

1.问题构建：

一个大型数据库的构建需要很长时间，为了减少数据需求方（科研人员）的等待时间，采取stream的形式，分批标注数据。由于不同批次的数据质量不一样，本文提出在这种stream场景下的学习方法。

2.本文贡献

1）提出了一种新方法解决分批标注数据的学习问题，并且新方法对于batch中的错误标注有很强的抵抗力；
2）本文是首先探索按照small-scale batch标注数据的工作；

3.论文的假设

本文研究一个较为新的领域，做了一些前提假设：
1）不同batch的数据质量不一样；
2）后续的数据质量比前面batch 的要高；
3）数据一旦标注，就不会回去重新更改标注了；

4.处理分批标注数据的三种传统方法

1）以往batch数据+新batch数据->当前数据，训练模型
2）$\text{batch}_1 \rightarrow \text{Model}_1, \text{batch}_2 + \text{Model}_1 + \text{finetune} \rightarrow \text{Model}_2 \ldots \rightarrow \text{Model}_n$
3）抛弃以往batch数据，新batch数据->当前数据，训练模型

5.本文方法

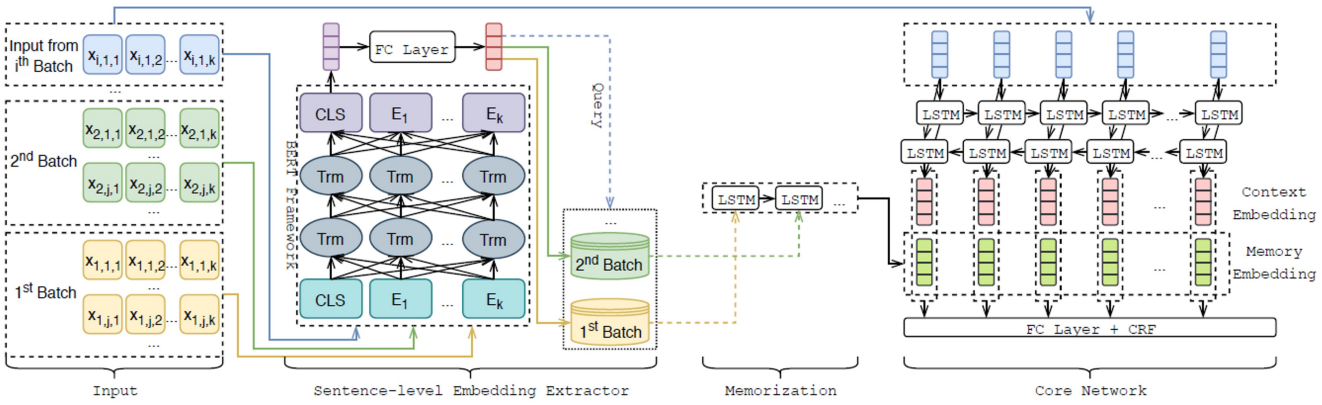

Figure 1: An overview of the proposed framework. For an input sentence in the i^{th} batch, it retrieves the most similar sentence-level embeddings from the previous batches – one embedding from one batch (Section 2.1). The sentence-level embeddings from previous batches go through a forward LSTM network which outputs memory embeddings (Section 2.2). Memory embeddings are then concatenated with the context embedding from the Bi-LSTM and fed into the CRF model. (Section 2.3).

Figure 1. 是本文方法的框架流程示意图。

通过预训练的BERT网络，将以往的batch数据{$batch_1, batch_2, \dots, batch_{i-1}$}佛重要的信息提取出来作为

6.实验

数据集：ACE2005 event annotation数据集上进行实验；

数据构造：模拟不同batch数据质量不一样，采取向batch中加入噪声标签，或者标签缺失的方法，让最近的batch数据质量比以前的batch数据质量好；为了研究本文方法的抗噪声能力，对比了两个噪声级别的实验：

1）25%: 25%, 10%, 5%, and 0%.

2）10%: 10%, 5%, 0%, and 0%.

实验结果：

从下面的表格可以看出，本文提出的方法效果很好；并且在噪声越严重的时候，本文方法的优势越明显；

Noise	Slice / Metric	200303 P	200303 R	200303 F1	200304 P	200304 R	200304 F1	200305 P	200305 R	200305 F1	200306 P	200306 R	200306 F1
25%	*All*	56.9	56.6	56.7	63.4	**68.7**	65.9	70.1	68.6	69.3	71.3	70.4	70.8
	Current	56.9	56.6	56.7	60.5	55.8	58.1	60.4	48.3	53.7	63.5	56.0	59.5
	Finetune	56.9	56.6	56.7	67.9	67.7	67.9	66.6	64.8	65.7	70.2	66.5	68.3
	Proposed	56.9	56.6	56.7	**74.2**	66.7	**70.3**	**71.2**	**74.1**	**72.7**	**72.9**	**76.1**	**74.4**
10%	*All*	67.9	70.4	69.1	80.1	78.3	79.2	82.7	**79.9**	81.2	83.4	82.3	82.8
	Current	67.9	70.4	69.1	64.8	60.8	62.7	54.4	57.2	55.8	63.5	56.0	59.5
	Finetune	67.9	70.4	69.1	72.7	75.6	74.1	77.3	74.8	76.0	74.5	76.7	75.5
	Proposed	67.9	70.4	69.1	**80.5**	**79.6**	**80.1**	**85.7**	78.9	**82.2**	**84.5**	**82.8**	**83.6**

Table 3: Performance (%) comparison with different strategies and noise level. Names of strategies (*All*, *Current*, *Finetune*, *Proposed*) and definition of noise level are introduced in Section 3.2. Note that there is no "previous" batch for the first batch, the results in the first batch are identical.

不同策略的计算时间开销对比：

Time	Training Time (Seconds)
All	18328
Current	12473
Finetune	12567
Proposed	14832

Table 4: Time consumption with different strategies.

本文方法比current和finetune开销多的地方在于：retrieving sentence-level embedding and training the memorization module.

7.未来工作：

在crowd-sourcing 的数据标注场景下，由于不同人员标注技术不同导致的不同batch数据质量波动很大，研究这种场景下的学习策略。

[1]https://arxiv.org/abs/2002.04165