

陈丹琦“简单到令人沮丧”的屠榜之作：关系抽取新SOTA！

原创 JayLou娄杰 夕小瑶的卖萌屋 2020-11-05 22:20

收录于话题

#卖萌屋@自然语言处理

69个



星标/置顶小屋，带你解锁
最萌最前沿的NLP、搜索与推荐技术

文 | JayLou娄杰

大家好，我是卖萌屋的JayJay，好久不见啦～

最近在「夕小瑶@知识图谱与信息抽取」群里和群友交流时，JayJay发现了来自陈丹琦大佬（女神）的一篇最新的关系抽取SOTA《**A Frustratingly Easy Approach for Joint Entity and Relation Extraction**》，光看题目就让人眼前一亮：是啥子简单方法，让实体关系的联合抽取方法“沮丧”了？

仔细阅读原文后，发现这篇paper采取pipeline方式就超越了一众联合抽取模型（joint model），登顶ACE04/05、SciERC榜首！

也许你会问：咦？现在的关系抽取SOTA不都是各种joint方式吗？有木有搞错？JayJay也有各种疑问：

1. 现在的关系抽取不是都采取joint方式、魔改各种Tag框架和解码方式吗？
2. 不是说pipeline方式存在误差积累吗，还会增加计算复杂度（实体冗余计算）吗？
3. 不是说pipeline方式存在交互缺失，忽略实体和关系两个任务之间的内在联系吗？
4. 不是说……

其实在这篇paper中，丹琦大佬并没有只是简单地刷刷SOTA而已，而就上述等若干问题进行了仔细探究～不过还是先po一下这篇SOTA的主要贡献和结论（开始正经，划重点）：

- 设计了一种非常简单的end2end关系抽取方法，即采取2个独立的编码器分别用于实体抽取和关系识别，使用相同的预训练模型就超越了之前所有的joint模型～
- 分别学习实体和关系的不同上下文表示，比联合学习它们更有效～
- 在关系模型的输入层融合实体类别信息十分重要～
- 提出了一种新颖并且有效的近似方法，在精度下降很小的情况下，就实现8-16倍的推断提速～

看到上述结论，也许我们需要打破这些刻板印象，重新审视所谓的“joint就一定好于pipeline”吧～下面赶紧和JayJay一起去膜（shen）拜（shi）一下这篇SOTA吧～

论文链接：

<https://arxiv.org/pdf/2010.12812.pdf>

Arxiv访问慢的小伙伴也可以在【夕小瑶的卖萌屋】订阅号后台回复关键词【1105】下载论文PDF～

这些年我们魔改过的joint模型

正式介绍本文的pipeline方法之前，我们先来回顾一下这些年我们魔改过的joint模型。实体关系的joint抽取模型可分为2大类：

第1类：多任务学习（共享参数的联合抽取模型）

多任务学习机制中，实体和关系共享同一个网络编码，但本质上仍然是采取pipeline的解码方式（故仍然存在误差传播问题）。近年来的大部分joint都采取这种共享参数的模式，集中在魔改各种**Tag**框架和解码方式。这里简单归纳几篇被大家熟知且容易实践的paper：

- **多头选择**^[1]：构建 $N \times N \times C$ 的关系分类器对每一个实体pair进行关系预测（N为序列长度，C为关系类别总数），输入的实体pair其实是每一个抽取实体的最后一个token。后续基于多头选择机制，也有paper引入预训练语言模型和bilinear分类。
- **层叠式指针标注**^[2]：将关系看作是SPO（Subject-Prediction-Object）抽取，先抽取主体Subject，然后对主体感知编码，最后通过层叠式的指针网络抽取关系及其对应的Object。
- **Span-level NER**^[3]：通过片段排列抽取实体，然后提取实体对进行关系分类。

第2类：结构化预测（联合解码的联合抽取模型）

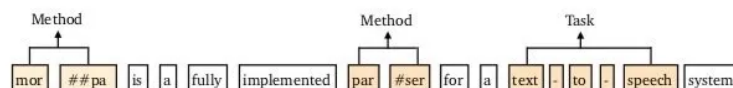
结构化预测则是一个全局优化问题，在推断的时候能够联合解码实体和关系（而不是像多任务学习那样，先抽取实体、再进行关系分类）。结构化预测的joint模型也有较多，比如统一的序列标注框架^[4]、多轮QA+强化学习^[5]等，不过有的联合解码方式通常很负责。

其实JayJay也有一段时间痴迷于各种joint魔改模型，如果大家有兴趣可以在知乎上直接搜索阅读JayJay的这篇文章《nlp中的实体关系抽取方法总结》。

如此简单的pipeline模型，居然可以登顶关系抽取SOTA

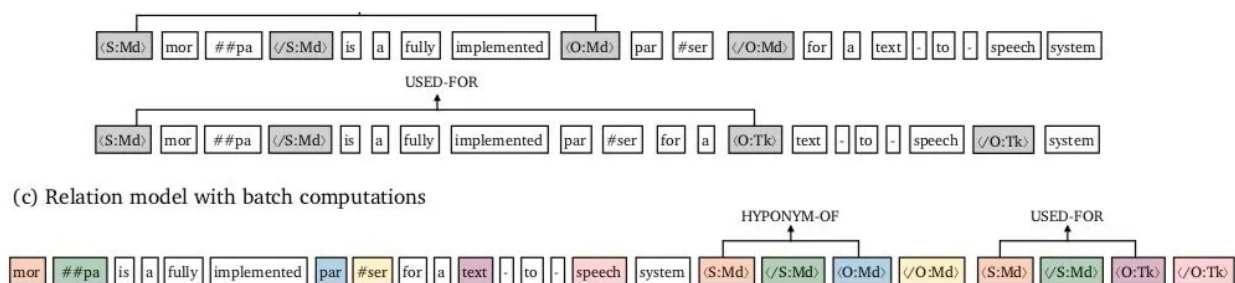
Input sentence:
MORPA is a fully implemented parser for a text-to-speech system.

(a) Entity model



(b) Relation model

HYPONYM-OF
↑



在这篇paper中，所采取的pipeline模型其实很简单：

- **实体模型**：如上图(a)所示，采取Span-level NER的方式，即基于片段排列的方式，提取所有可能的片段排列，通过SoftMax对每一个Span进行实体类型判断。这样做的好处是可以解决嵌套实体问题，但计算复杂度较高，因此需要限制Span长度（对于含 n 个token的文本，理论上共有 $n(n+1)/2$ 种片段排列）。
- **关系模型**：如上图(b)所示，对所有的实体pair进行关系分类。其中最重要的一点改进，就是将实体边界和类型作为标识符加入到实体Span前后，然后作为关系模型的input。例如，对于实体pair（Subject和Object）可分别在其对应的实体前后插入以下标识符：
 - **<S:Md>和</S:Md>**：代表实体类型为Method的Subject，S是实体span的第一个token，/S是最后一个token；
 - **<O:Md>和</O:Md>**：代表实体类型为Method的Object，O是实体span的第一个token，/O是最后一个token；

对于关系模型，对每个实体pair中第一个token的编码进行concatenate，然后进行SoftMax分类。

需要特别指出的是，上述实体模型和关系模型采取的两个独立的预训练模型进行编码（不共享参数）。

对于这种关系模型，我们不难发现：对每个实体pair都要轮流进行关系分类，也就是同一文本要进行多次编码（呃～，心累，计算开销必然会很大啊）。

为解决这一问题，提出了一种加速的**近似模型**（如上图c所示）：可将实体边界和类型的标识符放入到文本之后，然后与原文对应实体共享位置向量。上图中相同的颜色代表共享相同的位置向量。具体地，在attention层中，文本token只去attend文本token、不去attend标识符token，而标识符token可以attend原文token。综上，通过这种「近似模型」可以实现一次编码文本就可以判断所有实体pair间的关系。

此外，由于跨句信息可用于帮助预测实体类型和关系（尤其是在代词性提及中），所以还通过简单方式引入了跨句信息，即文本输入的左右上下文中分别滑动 $(W-n)/2$ 个words， n 为文本长度， W 为固定窗口大小。

Model	Encoder	ACE05			ACE04			SciERC		
		Ent	Rel	Rel+	Ent	Rel	Rel+	Ent	Rel	Rel+
(Li and Ji, 2014)	-	80.8	52.1	49.5	79.7	48.3	45.3	-	-	-
(Miwa and Bansal, 2016)	L	83.4	-	55.6	81.8	-	48.4	-	-	-
(Katiyar and Cardie, 2017)	L	82.6	55.9	53.6	79.6	49.3	45.7	-	-	-
(Zhang et al., 2017)	L	82.6	55.9	53.6	79.6	49.3	45.7	-	-	-

(Zhang et al., 2017a)	L	85.0	-	57.5	-	-	-	-	-	-
(Luan et al., 2018) ^{♣†}	L+E	-	-	-	-	-	-	64.2	39.3	-
(Luan et al., 2019) ^{♣†}	L+E	88.4	63.2	-	87.4	59.7	-	65.2	41.6	-
(Li et al., 2019)	BI	84.8	-	60.2	83.6	-	49.4	-	-	-
(Dixit and Al-Onaizan, 2019)	L+E	86.0	-	62.8	-	-	-	-	-	-
(Wadden et al., 2019) ^{♣†}	Bb	88.6	63.4	-	-	-	-	-	-	-
(Wadden et al., 2019) ^{♣†}	SciB	-	-	-	-	-	-	67.5	48.4	-
(Lin et al., 2020)	BI	88.8	67.5	-	-	-	-	-	-	-
(Wang and Lu, 2020)	ALB	89.5	67.6	64.3	88.6	63.3	59.6	-	-	-
Ours: single-sentence	Bb	88.7	67.0	62.2	88.1	62.8	58.3	-	-	-
	SciB	-	-	-	-	-	-	66.6	48.4	35.1
	ALB	89.7	69.0	65.6	-	-	-	-	-	-
Ours: cross-sentence [♣]	Bb	90.2	67.7	64.6	89.2	63.9	60.1	-	-	-
	SciB	-	-	-	-	-	-	68.2	50.1	36.7
	ALB	90.9	70.2	67.8	90.3	66.1	62.2	-	-	-

上图给出了各个关系抽取榜单的对比结果（梅花图标代表引入了跨句信息，Rel+为严格指标），可以看出：

1. 不使用跨句信息(单句设置)就超越了ACE04/ACE05曾经的SOTA（使用同一预训练模型），使用跨句信息更是了不得啊。
2. 使用跨句信息登顶了SciERC（文档级）的SOTA；

Model	ACE05		SciERC	
	Rel (F1)	Speed (sent/s)	Rel (F1)	Speed (sent/s)
Full (single)	67.0	32.1	48.4	34.6
Approx (single)	66.5	384.7	47.4	301.1
Full (cross)	67.7	14.7	50.1	19.9
Approx (cross)	66.8	237.6	48.8	194.7

而上图也给出了加速版「近似模型」的效果，可以看出：

1. 在单句设置中，ACE05和SciERC分别提速11.9倍和8.7倍，而指标仅仅下降0.5%和1.0%。
2. 「近似模型」加速明显（划重点）：我们先不要考虑是不是SOTA的事情，这个「近似模型」one-pass编码的方式就足够让我们眼前一亮，这种方式与预训练模型兼容的恰到好处，权衡性能和准确性，相信在实际应用中具有重要意义！

好了，到这里为止，pipeline模型就介绍完了；不知道你是否和JayJay一样充满疑问：如此简单的pipeline模型，竟然可以超越那些曾经登顶SOTA的众多joint模型，**WHY?**

pipeline如此强悍，自有它的道理，也有“先兆”

pipeline为何如此强悍，是哪些因素让它可以“秒杀”众多joint模型？在这一部分我们以QA的形式逐一去揭晓（PS：最近几篇顶会paper中的论点，也和这篇SOTA互相佐证，原来早有先兆啊～）。

Q1、关系抽取最care什么？实体类型信息也太重要了吧

关系抽取最care什么？论文对关系模型起关键作用的因素进行了探究，首先就是分析不同实体表征方式的影响。文中共建了6种实体表征方式：

- **TEXT**：直接提取原始文本中，实体span所对应的编码表示。
- **TEXTETYPE**：在**TEXT**的基础上，concatenate实体类别向量。
- **MARKERS**：将标识符**S**、**/S**、**O**、**/O**插入到原始文本中，但是标识符没有实体类别信息。
- **MARKERSETYPE**：在**MARKERS**的基础上，concatenate实体类别向量，这是一种隐式的融入实体类别的方法。
- **MARKERSELOSS**：在关系模型中，构建判别实体类别的辅助loss。
- **TYPEDMARKERS**：就是本文所采取的方法，实体类别“显式”地插入到文本input中，如<S:Md>和</S:Md>、<O:Md>和</O:Md>。

Features	ACE05 (gold)	SciERC (gold)
TEXT	67.6	61.7
TEXTETYPE	68.2	63.6
MARKERS	70.5	68.2
MARKERSETYPE	71.9	68.9
MARKERSELOSS	70.7	68.0
TYPEDMARKERS	73.1	69.1

如上图所示（关系模型的实体是gold输入），我们可以发现：

1. 本文采取的**TYPEDMARKERS**实体表征方式，秒杀其余方式；
2. 实体类别信息对关系模型很重要，“显式”地编码实体类别信息好于隐式编码；在构造标识符时，不仅要区分span边界、更要显示融入实体类别信息。
3. **MARKERSETYPE**比**MARKERSELOSS**好，说明直接引入实体类型特征好于辅助loss的设计。

JayJay发现：这篇SOTA并不是第一次将实体标识符加入到文本input中，去年在Google的论文**Matching the Blanks**^[6]中就采取标识符强化实体表征，但却没有引入实体类型进行标识，只是采取**MARKERS**那种方式标记。

JayJay还发现：实体类别信息对于关系模型很重要这一结论，在最新的EMNLP20的一篇刘知远老师团队的《**Learning from Context or Names? An Empirical Study on Neural Relation Extraction**》中也被提及和证明。

anyway，记住一点：引入实体类别信息会让你的关系模型有提升～

Q2、共享编码 VS 独立编码 哪家强？

在JayJay的刻板印象中：基于参数共享编码的joint模型能够建模实体和关系的交互、促进彼此。但这篇居然是采取两个独立的编码器分别构建实体模型和关系模型，WHY？

不过，论文也给出了两个模型共享编码进行联合优化的实验，如下图所示：

Shared encoders?	✗	✓
Entity F1	88.8	87.7
Relation F1	64.8	64.4

可以看出共享编码反而使实体和关系的F1都下降了，丹琦大佬也解释道：这是由于两个任务各自是不同的输入形式，并且需要不同的特征去进行实体和关系预测，也就是说：**使用单独的编码器确实可以学习更好的特定任务特征。**

不过，JayJay认为：**不能一概而论地就认为独立编码就一定好于共享编码**，或许是共享编码机制过于简单了呢？但不可否认，对于实体和关系确实需要特定的特征编码，在构建joint模型时如果只是简单的强行共享编码，真的可能会适得其反。

真是很巧，最近EMNLP20的一篇《**Two are Better than One: Joint Entity and Relation Extraction with Table-Sequence Encoders**》（这篇SOTA之前的SOTA）也认为：大多数joint模型都是共享同一个编码，但这种方式存在一个问题：**针对一项任务提取的特征可能与针对另一项任务提取的特征一致或冲突，从而使学习模型混乱**。于是乎，作者设计了2种不同的编码器（表格编码器和序列编码器），以在表示学习过程中互相帮助，最后通过实验表明使用2个编码器好于只使用1个编码。

anyway，记住一点：**对于实体关系抽取，2个独立的编码器也许会更好～**

Q3、误差传播不可避免？还是不存在？

众所周知，pipeline不是存在「误差传播」吗？也就是说，关系模型在训练的时候，输入的是gold实体进行关系判断，这会导致训练和推断时候的差异（暴露偏差）。

那是不是在训练的时候输入预测的实体进行关系判断，会在推断时效果变好呢？于是论文采用10-way jackknifing方式（一种交叉验证）做了相关实验，发现训练时输入预测实体反而降低了F1值，毕竟这种方式会在训练的时候引入噪音啊～

采取pipeline进行推断时，如果有些gold实体在实体模型中没有被抽取出来，那么关系模型也就不会预测与该实体相关联的任何关系了。那有没有可能通过召回更多的实体，来减轻这种误差传播呢？论文也尝试召回更多的实体进行了实验，发现并没有提升F1值。

通过上述实验发现一些尝试均未显著改善性能，而论文提出的简单的pipeline却证明是一种出乎意料的有效策略～但丹琦大佬也指出：**并不认为误差传播问题不存在或无法解决，而需要探索更好的解决方案来解决此问题。**

不过，JayJay认为：pipeline存在误差传播，而那些基于共享编码的joint模型也存在误差传播啊，至于是不是真的会减轻这种误差传播也有待考证。

anyway，留给我们的一个课题就是：**误差传播怎么解决？**

还好，我们NLPer一直在尝试。最近COLING2020的一篇paper^[7]为了缓解这个问题，提出了一种单阶段的联合提取模型**TPLinker**，其不包含任何相互依赖的抽取步骤，因此避免了在训练时依赖于gold的情况，从而实现了训练和测试的一致性。是不是很神奇？感兴趣的小伙伴，赶快去阅读吧～

你打破“joint好于pipeline”的刻板印象了吗？

读完这篇SOTA，也许我们不会再有“joint好于pipeline”的感脚了。但这并不是说，joint就比不上pipeline了，或许我们未来可以设计出更好的joint框架还是会登顶SOTA。

对于JayJay来说，是不是SOTA其实没有那么重要～更重要的是，我们可以进一步尝试或验证这些trick：

- 引入实体类别信息会让你的关系模型有提升～
- 对于实体关系抽取，2个独立的编码器也许会更好～

当然还有一些需要我们进一步思考或解决的问题：

- 怎么减轻/避免误差传播呢？
- pipeline是不是推断效率还是慢？比如，还是会存在冗余实体的计算？
- 落地场景中，是不是pipeline更容易迭代优化呢？
- 下一个重新登顶SOTA的joint模型长啥样子啊？

JayJay不说再见，下一篇继续关注信息抽取～



萌屋作者：JayJay

知乎ID：JayLou娄杰。混迹于工业界的NLP算法工程师，擅长信息抽取和低资源NLP方向，聚焦于医疗数据深度结构化。

开源项目DeepIE：基于深度学习的信息抽取技术📌

<https://github.com/loujie0822/DeepIE>

作品推荐：

1. [NLP中的少样本困境问题探究](#)
2. [工业界求解NER问题的12条黄金法则](#)
3. [别让数据坑了你！用置信学习找出错误标注（附开源实现）](#)
4. [FLAT：中文NER屠榜之作！](#)
5. [打破BERT天花板：11种花式炼丹术刷爆NLP分类SOTA！](#)



后台回复关键词【入群】

加入卖萌屋NLP/IR/Rec与求职讨论群

有顶会审稿人、大厂研究员、知乎大V和妹纸

等你来撩哦~



参考文献

- [1] Joint entity recognition and relation extraction as a multi-head selection problem:
<https://www.sciencedirect.com/science/article/abs/pii/S095741741830455X?via%3Dihub>
- [2] A Novel Cascade Binary Tagging Framework for Relational Triple Extraction:
<https://arxiv.org/pdf/1909.03227.pdf>
- [3] Span-based Joint Entity and Relation Extraction with Transformer Pre-training:
<https://arxiv.org/pdf/1909.07755.pdf>

[4]Joint extraction of entities and relations based on a novel tagging scheme:

<https://arxiv.org/pdf/1706.05075.pdf>

[5]Entity-Relation Extraction as Multi-Turn Question Answering : <https://arxiv.org/pdf/1905.05529.pdf>

[6]Matching the Blanks: Distributional Similarity for Relation Learning: <https://arxiv.org/pdf/1906.03158.pdf>

[7]TPLinker: Single-stage Joint Extraction of Entities and Relations Through Token Pair Linking:

<https://arxiv.org/pdf/2010.13415.pdf>

喜欢此内容的人还喜欢

我不看好data2vec这类多模态融合的研究

夕小瑶的卖萌屋