# 1 实验目的

本实验目的为实现目标文本的事件抽取，即给定目标事件类型和角色类型集合，识别输入文本中包含的所有事件和角色，其中事件的识别需要识别事件类型（event type），角色的识别需要识别角色类别（role）和论元短语（arguments）。

数据集来自百度2020语言与智能技术竞赛，数据集包含65种事件类型，每个事件类型可能包含2-6种不同的角色。事件类型根据百度风云榜的热点榜单选取确定，具有较强的代表性。65个事件类型中不仅包含「结婚、辞职、地震」等传统事件抽取评测中常见的事件类型，还包含了「点赞」等极具时代特征的事件类型。数据集中的句子来自百度信息流资讯文本，相比传统的新闻资讯，文本表达自由度更高。数据集规模为：训练集11958条，测试集1498条。本次实验在训练集上进行训练和交叉验证，在测试集上评价模型的性能，并对实验结果进行详细分析。

数据集样例如下所示：



评价指标：事件识别的精准率（precision），召回率（recall）和 F1 值。角色识别的精准率（precision），召回率（recall）和 F1 值（要求角色类别和短语的严格匹配）。

# 2 实验原理

中文事件抽取方法主要分为基于模式匹配的事件抽取和基于机器学习的事件抽取两类[1]。

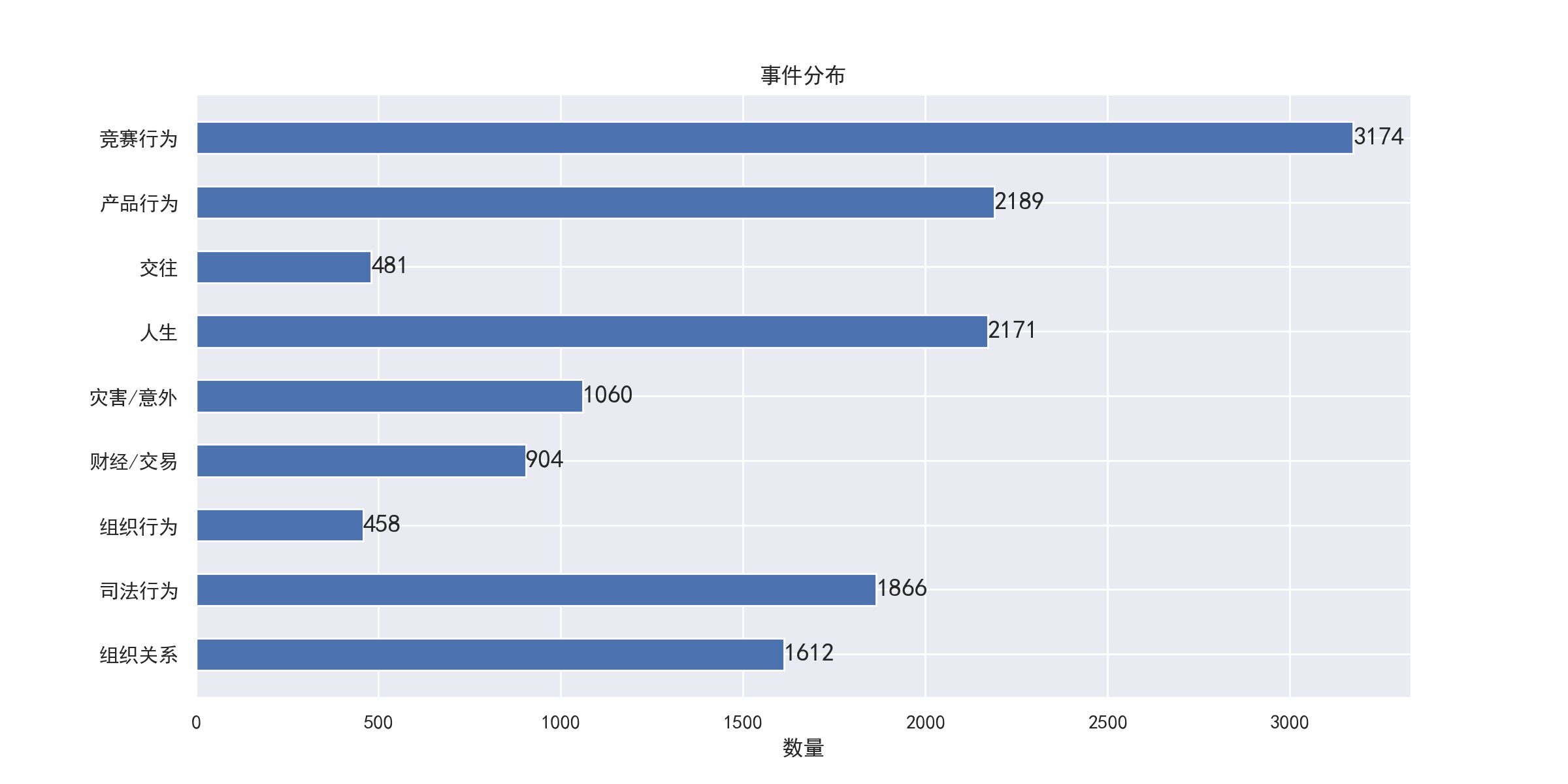
模式匹配在特定领域的性能较好，其核心在于模式的获取，需要基于领域知识通过手工构建的方式获得，移植性差。使用机器学习方法的事件抽取只需要训练语料，不需要太多的专家知识，移植性好，已成为目前事件抽取的主流。传统的机器学习方法通常先对语料进行预处理，如词性标注、命名实体识别、句法分析等，需要大量的特征选择。

随着深度神经网络的发展，深度学习方法可以自动学习特征，使数据具有更好的特征表达。预训练模型如Bert[2]、ENRIE[3]、XLNet[4]等，从大规模的语料中学习到字符级、词级、句子级甚至句间关系的特征，得到通用的语言表示，迁移到如机器阅读理解、文本分类、自然语言生成等下游学习任务，效果通常优于不使用预训练模型的方法。本实验报告将在第4节实验内容部分介绍中文事件抽取任务主要思路和流程。

# 3 数据集分析

数据集包含9类事件，65种事件类型，每个事件类型可能包含2-6种不同的角色。以下数据分析全部基于训练集。

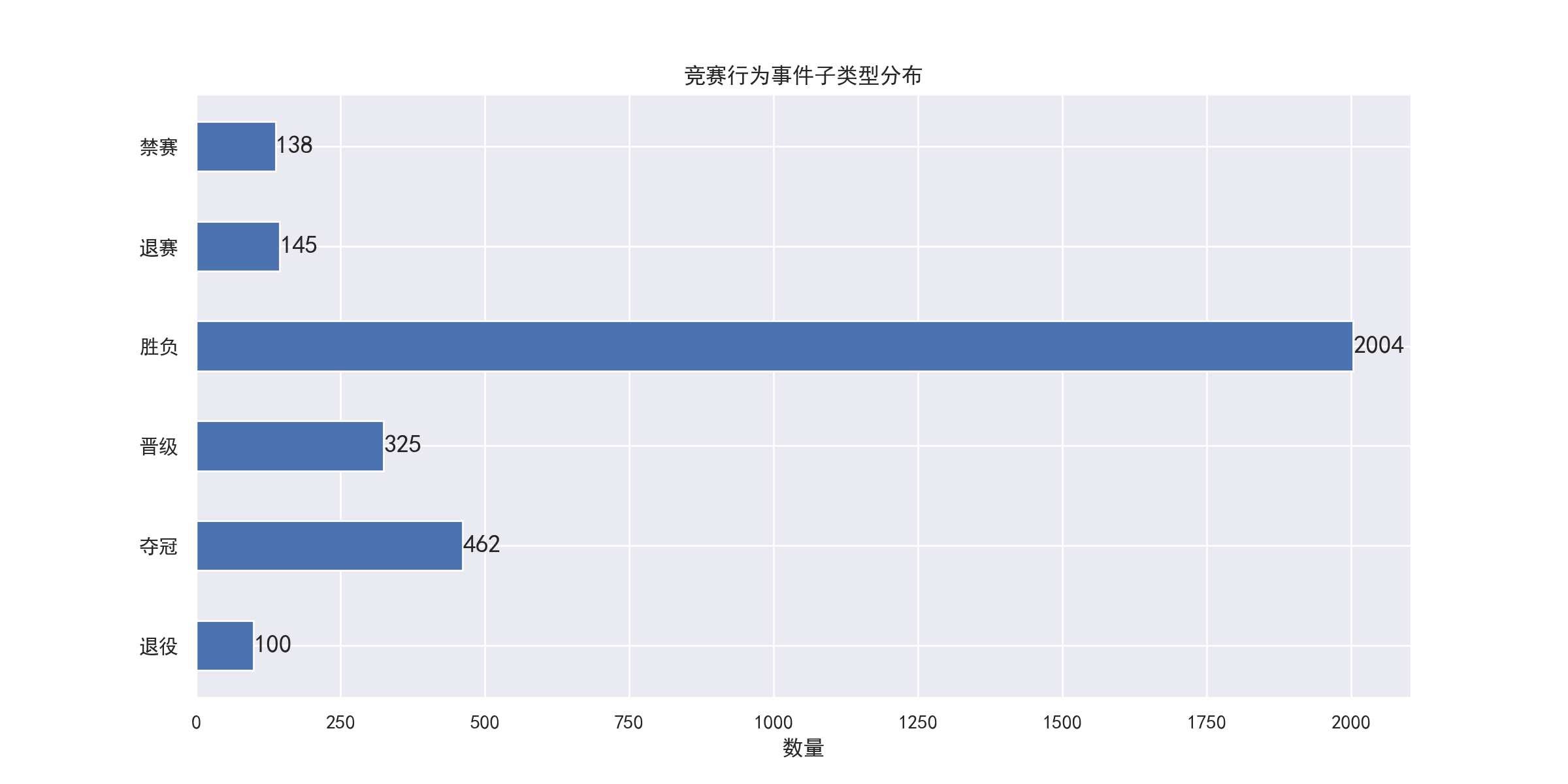
## 3.1 事件（class）



「竞赛行为」事件数量较多，「组织行为」、「交往」事件数量较少。

## 3.2 事件类型（event type）

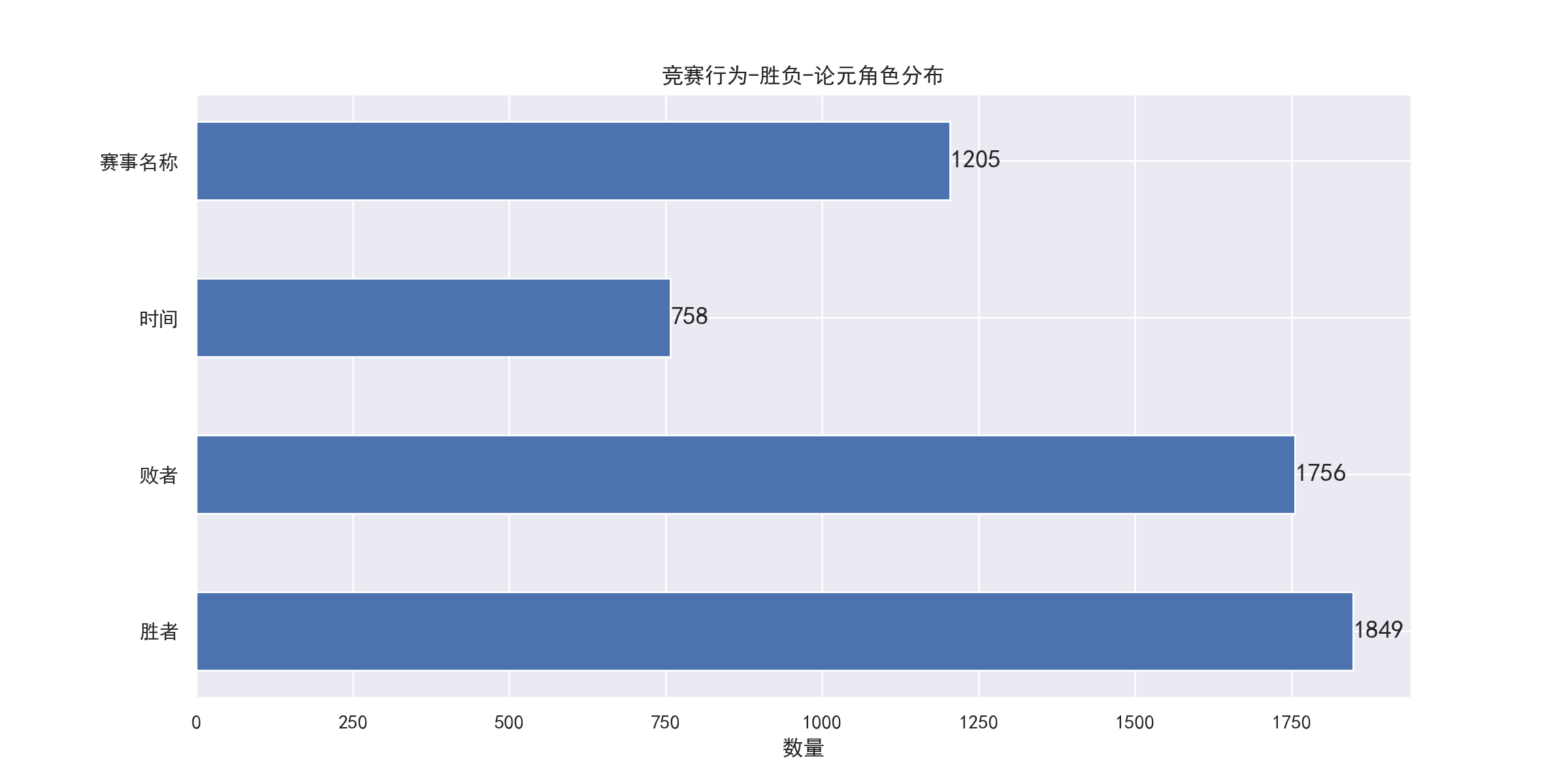
以「竞赛行为」事件为例，它包含：夺冠、晋级、禁赛、胜负、退赛、退役，一共6个事件类型。



「胜负」这一事件类型数量最多，「禁赛」、「退赛」、「退役」较少，各事件类型的数量不均衡。

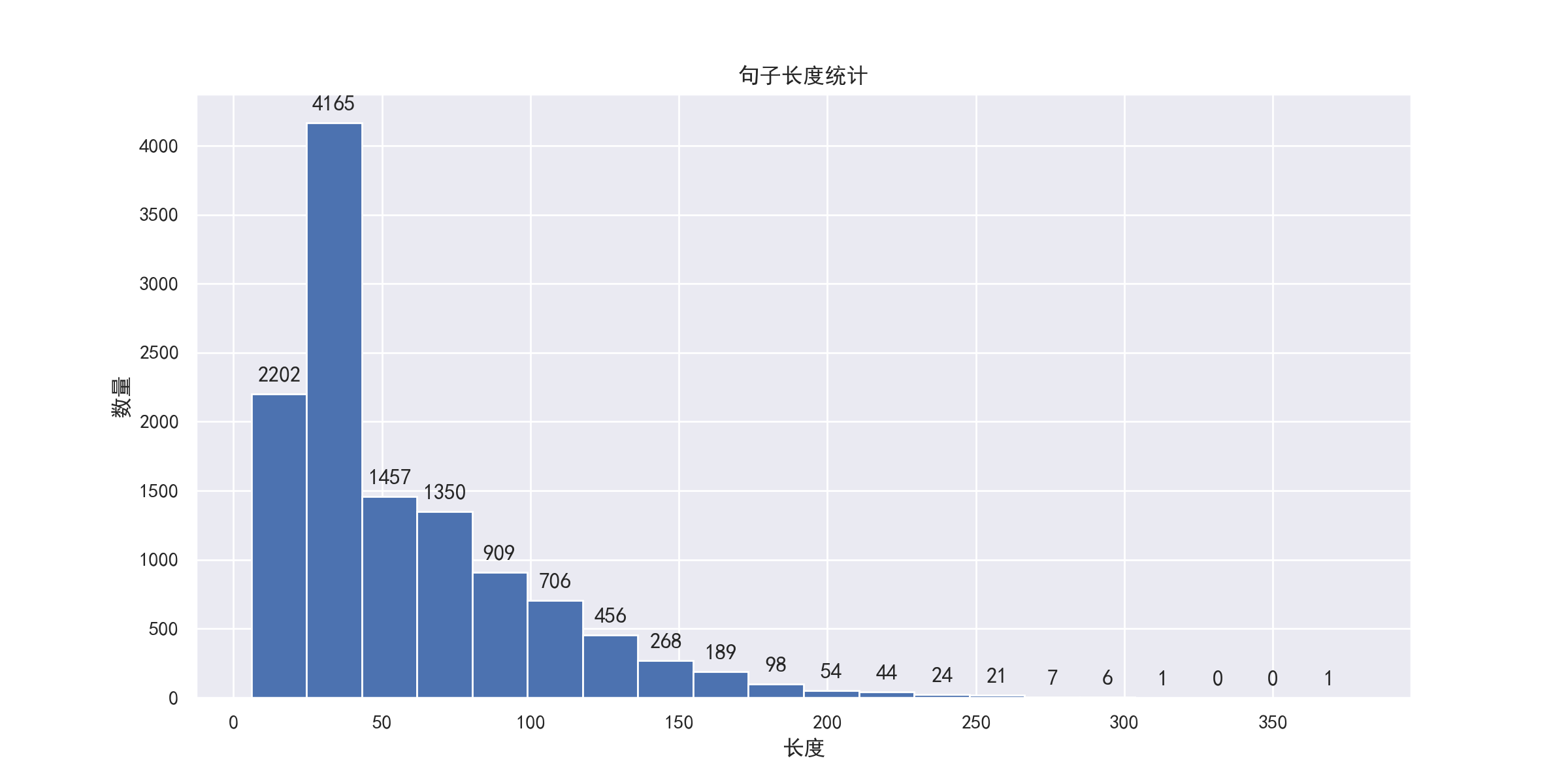
## 3.3 论元角色

「竞赛行为-胜负」这一事件类型中包含：时间、胜者、败者、赛事名称，一共4种论元角色。「胜者」、「败者」两种角色数量较多，「时间」较少。



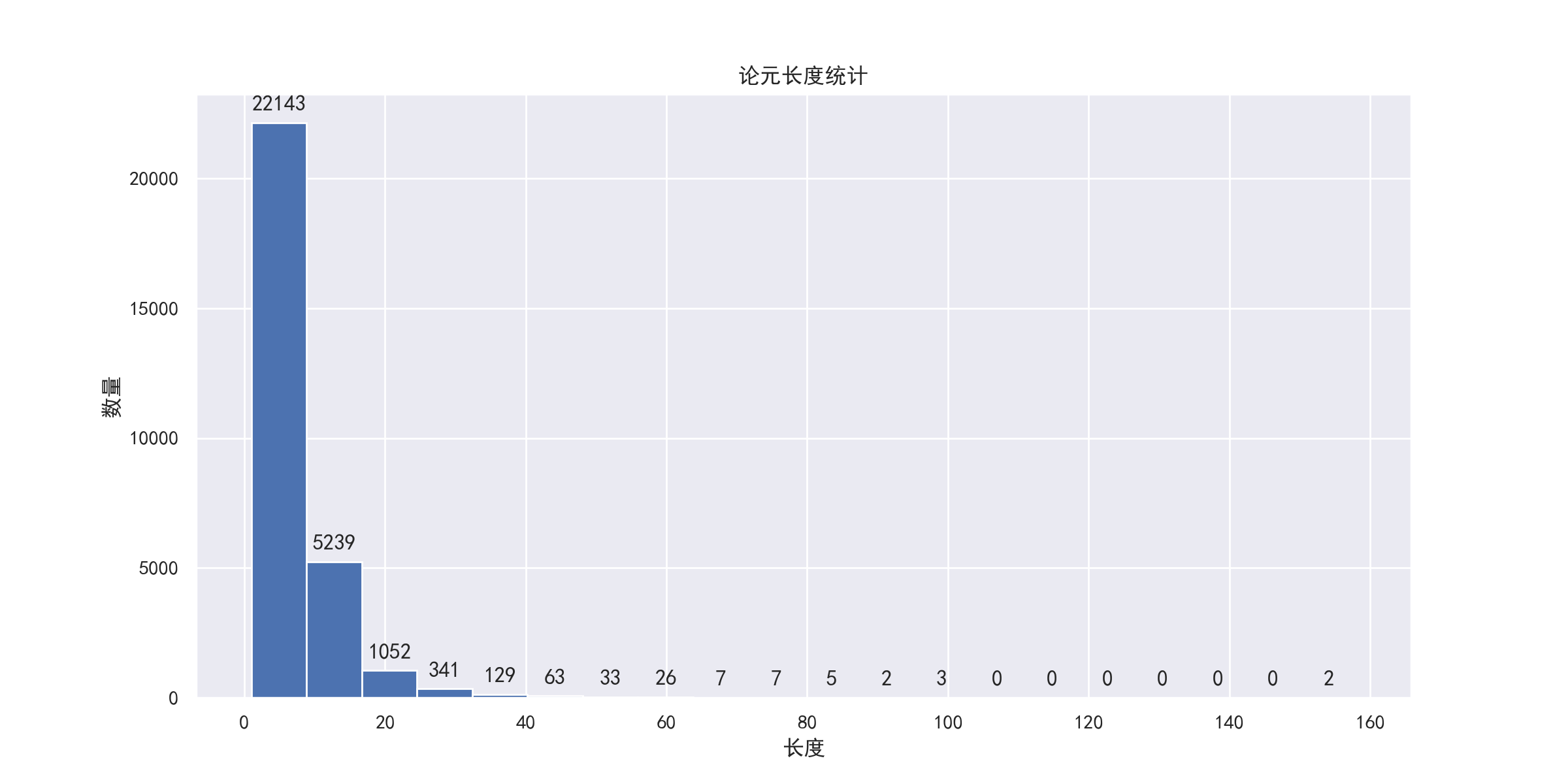
## 3.4 句子长度

占比95.85%的句子长度小于150。最短、最长的句子长度分别为：6，378。



## 3.5 论元长度

占比98.07%论元短语长度小于25。最短和最长论元长度分别为1，158。



## 3.6 句子中事件数量

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 事件数量 | 1 | 2 | 3 | 4 | >=5 | 总计 |
| 句子数量 | 10400 | 1292 | 193 | 46 | 27 | 11958 |
| 百分比（%） | 86.97 | 10.80 | 1.61 | 0.38 | 0.23 | 100 |

占比约87%的句子只包含一个事件，约11%句子包含两个事件。

## 3.7 句子中角色数量

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 角色数量 | 0 | 1 | 2 | 3 | 4 | 5~10 | >=11 | 总计 |
| 句子数量 | 305 | 2863 | 4507 | 2459 | 968 | 794 | 62 | 11958 |
| 百分比（%） | 2.55 | 23.94 | 37.69 | 20.56 | 8.09 | 6.64 | 0.52 | 100 |

占比约2.55%的句子中没有论元角色。约82%包含1-3个角色，占比最多。约15%的句子中角色数量大于3。

## 3.8 角色重叠（角色共享）

第1小节数据集样例中，

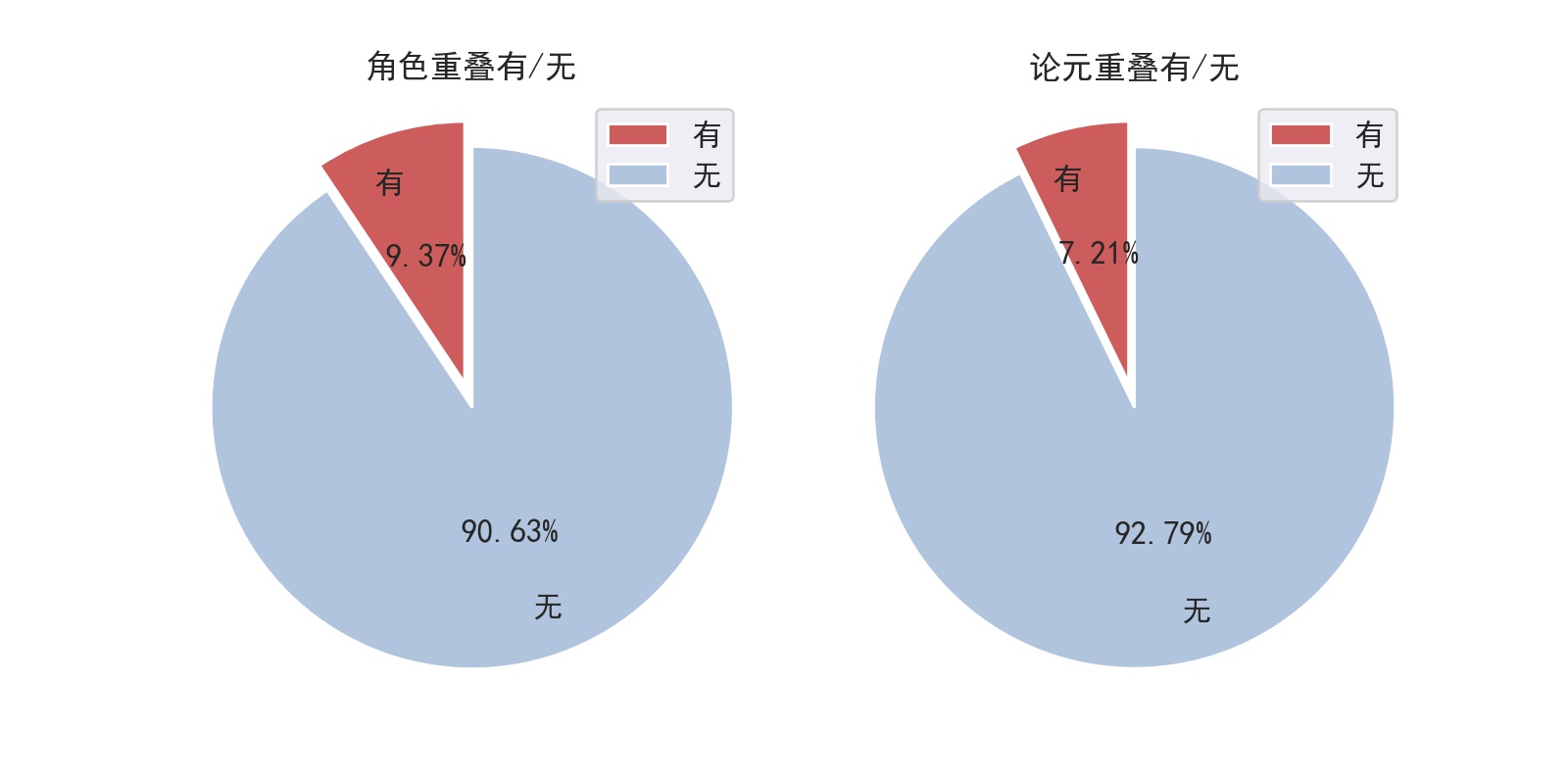
text: "北京时间昨天，南美足联决定：因为阿根廷球星梅西在今年美洲杯赛期间指责裁判腐败的言论，对他给予禁止随阿根廷队参加国际赛事3个月的处罚。根据这一决定，他将无法参加今年的4场友谊赛。与此同时，南美足联还对梅西罚款5万美元。"  
司法行为-罚款：（罚款对象，梅西）  
竞赛行为-禁赛：（被禁赛人员，梅西）

「梅西」这一论元，作为两个不同的角色出现在同一个句子中，这一现象称为角色重叠（角色共享）。有1120个样本存在这一现象，占比9.37%。

## 3.9 论元重叠

text: "也门国防部29日晚发表声明说，阿联酋战机自28日晚起对驻扎在也门南部多个地区的政府军发动10次空袭，导致300多人死伤"  
灾害/意外-袭击：（袭击对象，驻扎在也门南部多个地区的政府军）  
灾害/意外-袭击：（地点，也门南部多个地区）

袭击对象和地点两个角色，均包含「也门南部多个地区」论元，这一现象称为论元重叠。有862个样本存在这一现象，占比7.21%。

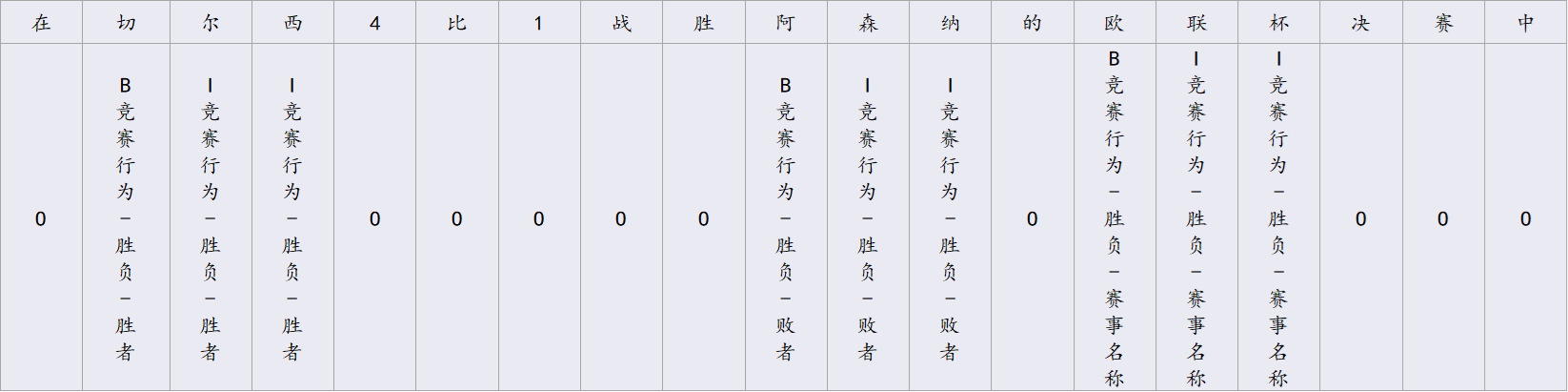


# 4 实验内容

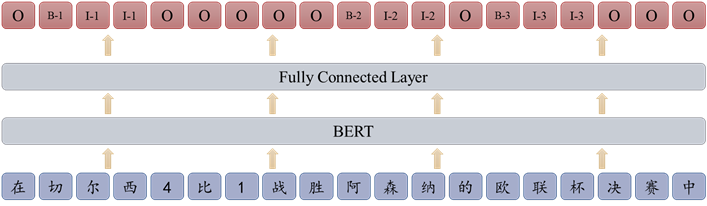
## 4.1 基于序列标注的模型

序列标注（sequence labeling）是分类问题的推广，在词性标注、命名实体识别[5]等任务中被广泛使用。序列标注模型的输入是观测序列，输出是标记序列，模型学习从输入到输出的映射关系。解决序列标注问题可分为概率图模型和深度学习模型。本实验使用后者。

基于序列标注的模型，其思路是给文本中的每个字打标签。给定的事件类型和论元角色组合，生成组合标签。如「竞赛行为-胜负」这一事件类型中包含：时间、胜者、败者、赛事名称，一共4种论元角色，可生成4种组合标签：「竞赛行为-胜负-时间」、「竞赛行为-胜负-胜者」、「竞赛行为-胜负-败者」和「竞赛行为-胜负-赛事名称」。本实验使用B/I/O标注方式对数据进行处理，示例如下：



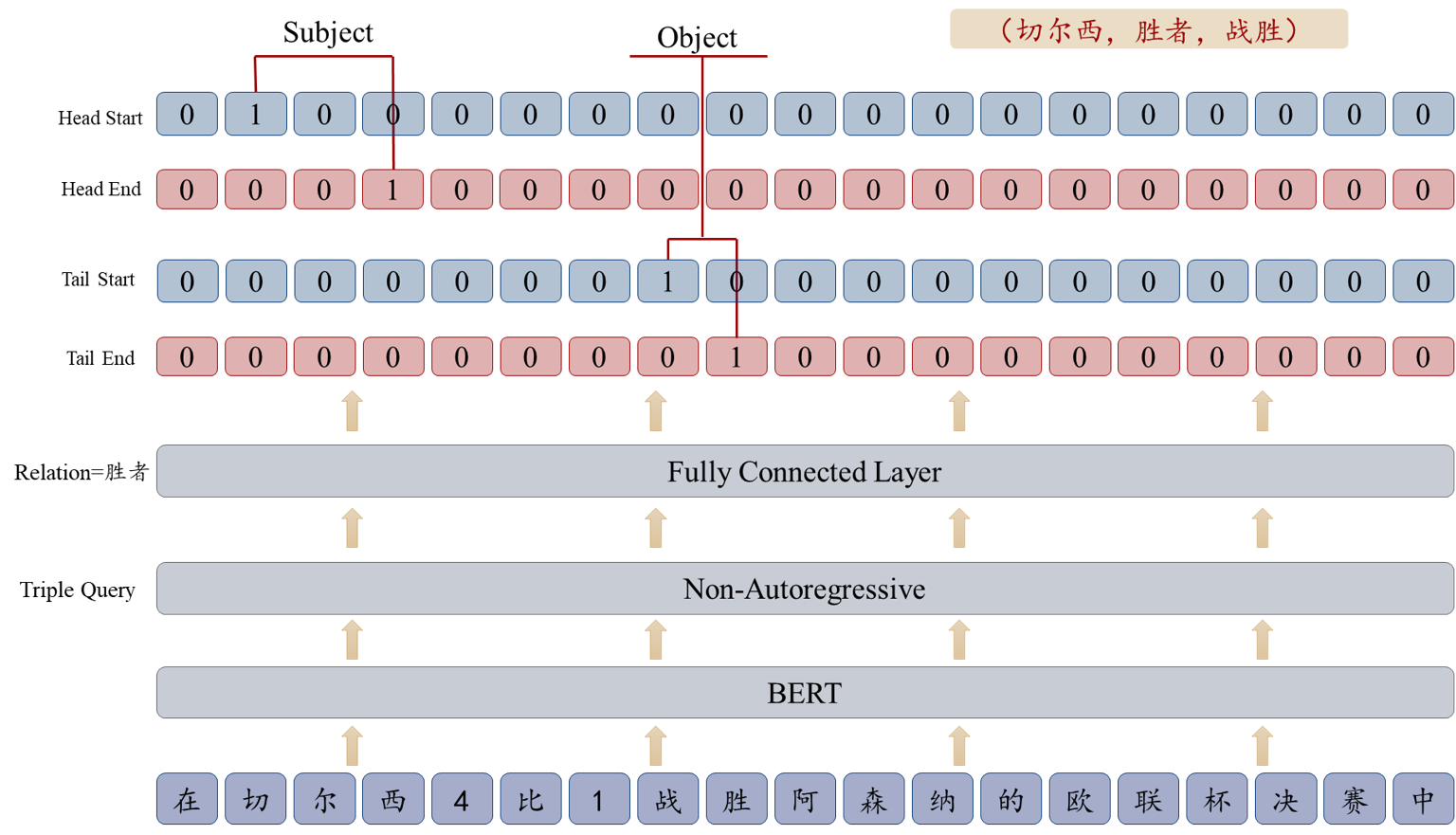
一共有65种事件类型，每种事件对应2-6种角色，组合标签种类数量为217，「O」表示不属于任何一个标签，总标签数量为217\*2+1=435。假定「竞赛行为-胜负-胜者」、「竞赛行为-胜负-败者」和「竞赛行为-胜负-赛事名称」三个组合标签对应序号分别为1、2和3。模型的结构如下图所示：



输入数据经过 BERT 编码层进行特征抽取，输出的特征向量进入全连接层进行解码，得到最终序列标注。由序列标注的结果即可确定事件类型和抽取论元短语。

## 4.2 基于三元组关系抽取的模型

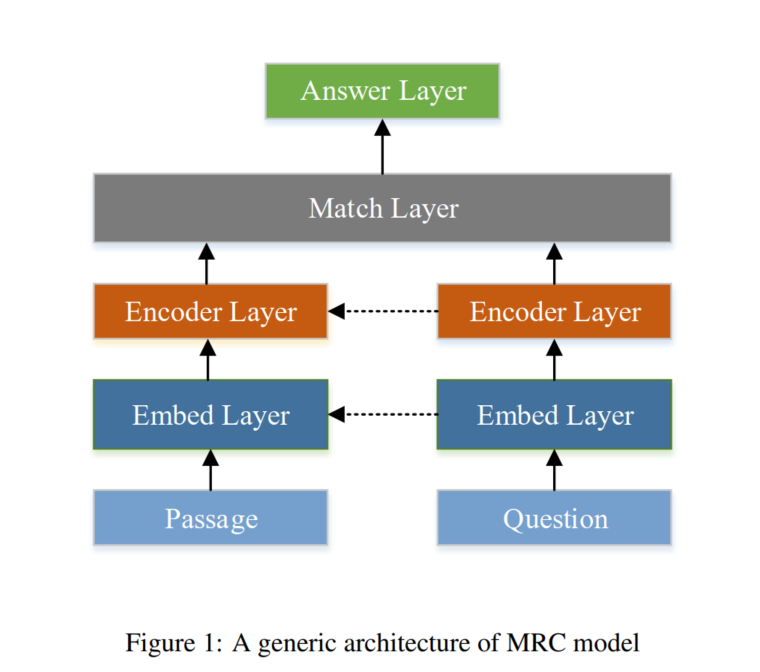
三元组关系抽取的目的是从给定一段文本中，识别所有形如（subject, relation, object），简写为（s, r, o）的三元组，其中 s 和 o 是文本中的实体，r 是给定模板中的关系类别。通常有四种关系抽取模式：1）pipeline，先识别实体，再识别实体间关系；2）填表法，实体和关系共享参数；3）序列标注；4）seq2seq，直接生成有某种关系的实体对[14]。

将三元组关系抽取应用于论元识别，可将论元短语（argument）看作是 s，将论元角色（role）看作是 r，将触发词（trigger）看作是 o。于是，抽取（s, r, o）转化为抽取（argument, role, trigger）形式的三元组。模型结构如下图所示：

输入数据经过 BERT 编码进行特征抽取，输出的特征向量进入非自回归层，结合三元组询问信息后经过全连接层解码，输出候选三元组。其中解码层使用自注意力建模各三元组的联系，注意力融合句子文本的信息，使用二部图匹配损失函数。该模型属于 one-step 模型，实现同时抽取论元短语和所属角色。

## 4.3 基于机器阅读理解的模型

机器阅读理解（Machine Reading Comprehension, MRC）是一种利用算法是计算机理解文章语义并回答相关自然语言问题的技术。MRC 的基本框架如下图所示[6]，通常 MRC 任务分为四种类型：完形填空、多选、区间抽取和自由回答。



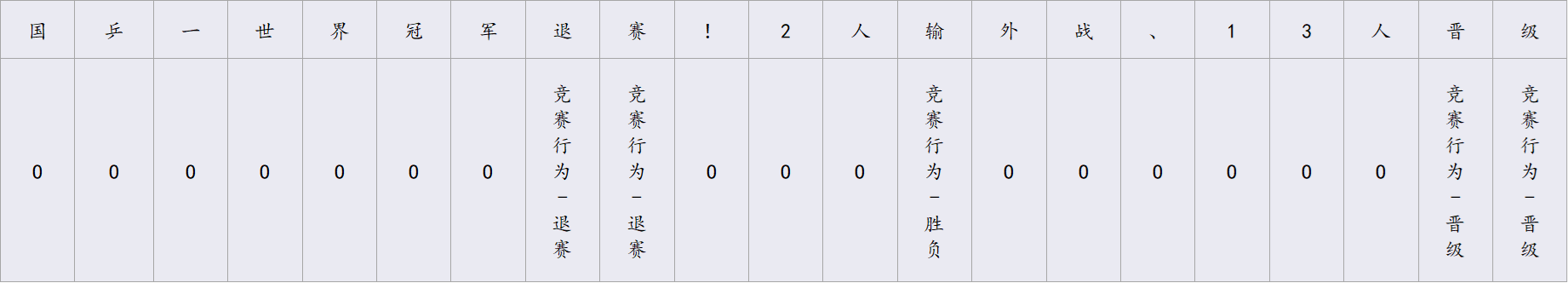
用机器阅读理解的方式实现事件抽取，属于区间（span）抽取类型。给定文本（passage）和问题（question），需要从 passage 中提取出 answer。将给定的句子视为 passage，根据目标设计 question，模型预测输出 answer 作为论元短语。这一方式的优点在于使用了论元角色信息；而第4.1节序列标注方式中，论元角色只当作标签进行分类，忽视了论元角色标签本身的语义信息，以及角色和论元短语之间的关系。文献[7]将 MRC 方式应用于命名实体识别任务，能有效识别嵌套实体。文献[10]将 MRC 应用于事件抽取，仅用 1% 的数据 F1 值达到49.8%，在 zero-shot 事件抽取场景中 F1值达到 37.0%。

本实验设计将事件类型与角色标签拼接作为 question，针对每种角色设计各自的问题。这样设计可以解决多个事件类型的识别、角色重叠和论元重叠问题。

实验思路：先识别事件类型，得到预测的事件类型后，再与该事件类型下的所有角色拼接，得到所有待询问的 question，最后使用 MRC 方式进行回答，抽取论元短语。此设计属于 pipeline 结构，会产生级联误差。

### 4.3.1 识别事件类型

使用文本中触发词信息，预测事件类型[8]。使用I/O标注方式，对触发词进行序列标注。文本中每一个触发词对应一种事件类型。如下表所示：

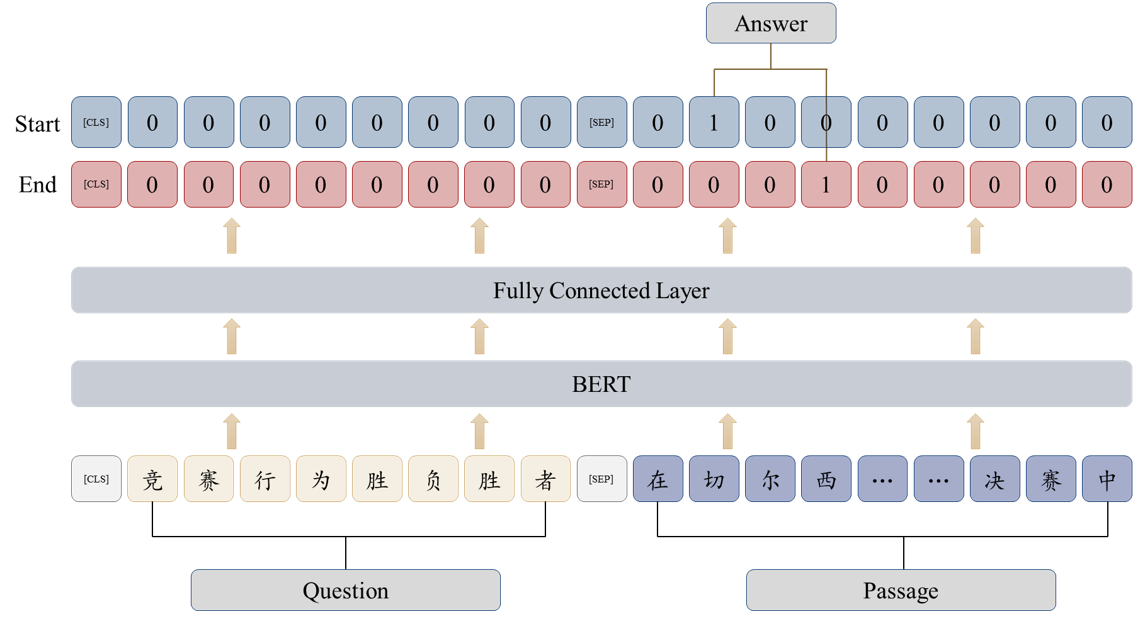


在上例中，「退赛」、「输」和「晋级」分别是事件类型「竞赛行为-退赛」、「竞赛行为-胜负」和「竞赛行为-晋级」的触发词。非触发词标注为「O」。一共有65种事件类型，标签数量为66。使用和第4.1节序列标注类似的模型，BERT+Fully Connected Layer，输入数据经过 BERT 编码层进行特征抽取，输出的特征向量进入全连接层解码，得到最终序列标注结果，即事件类型标签。

### 4.3.2 MRC 抽取论元

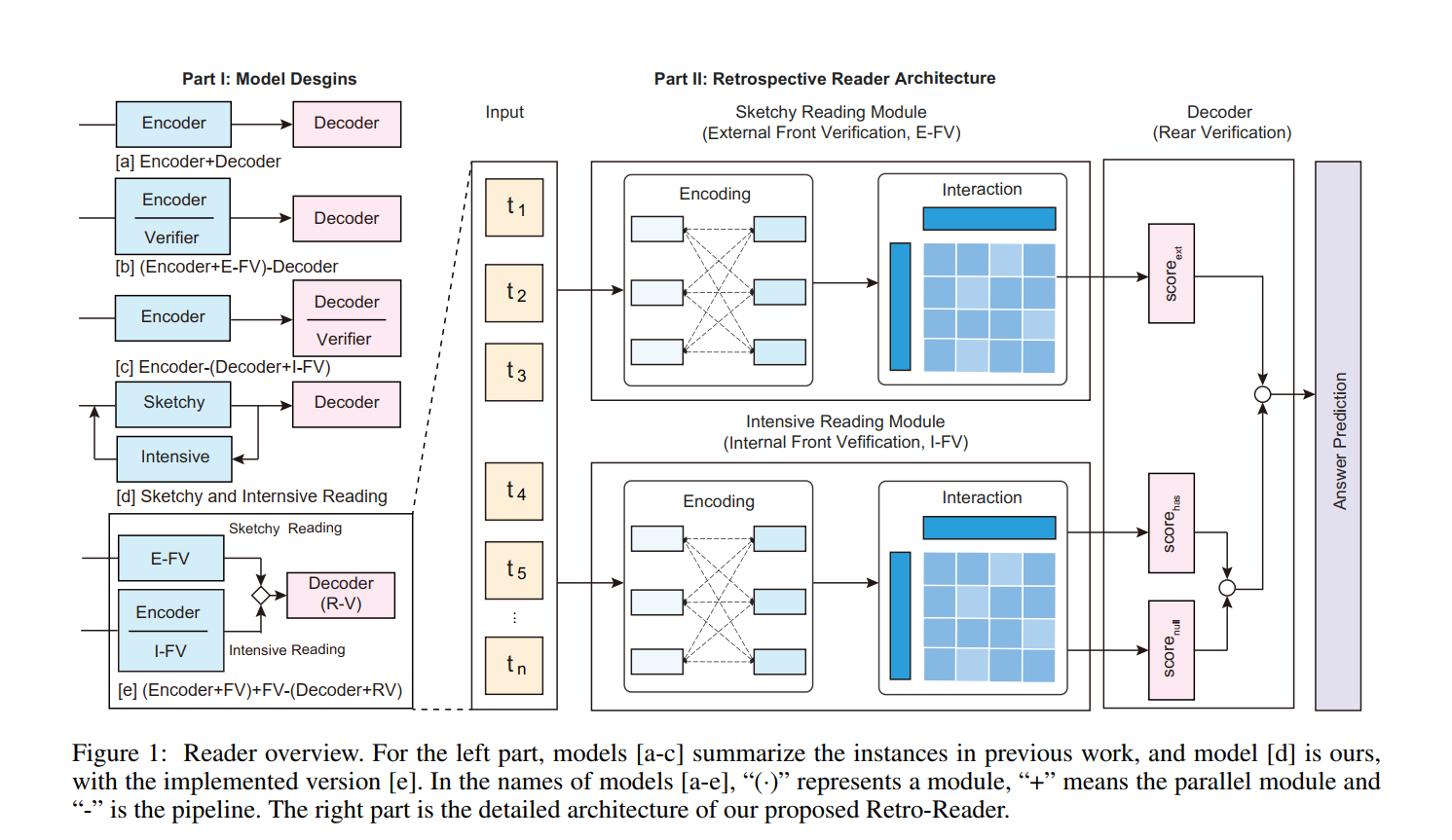
设计 question 为事件类型和角色标签组合，passage 为文本序列，待预测的 answer 为 passage 中一个或多个连续 span。关于 span 抽取有两种方式：标注 span 开始和结束位置[2]，标注 span 全部字所在位置[9]。

若文本中不包含某一角色，那么该问题的答案为空，即没有可抽取的论元作为答案。模型训练时使用真实的事件类型，测试时使用预测的事件类型。以标注 span 开始和结束位置为例，模型结构如下图所示：



拼接 question 和 passage 作为输入数据，经过 BERT 编码层进行特征抽取，输出的特征向量进入全连接层进行解码，输出每个字作为 start / end 的概率。提取所有距离匹配最近的 start 和 end 所在位置里的所有字符，得到询问 question 的 answer，即实现论元短语抽取。

若采用标注 span 全部字所在位置的方式，则模型输出每个字是否属于答案的概率。提取所有分类为属于答案的字符，实现论元短语抽取。



从实验结果中发现约有 70% 的误差来源于使用 MRC 方式抽取论元，预测问题有无答案时出现错误，即不能很好的区分文本中是否存在对应论元角色的论元短语。预测并回答了真实标签为无答案的问题，也将真实标签为有答案的问题预测为无答案。

文献[11]、[12]中提到类比人类做阅读理解的方式，将 MRC 抽取论元分为两个阶段。第一阶段略读（sketchy reader），判断询问的问题是否可回答。第二阶段精读（intensive reader）同时预测答案和判断是否可回答。最后综合验证（rear verification）略读、精读两个阶段，输出最终结果：判断问题是否可回答，若可则给出预测结果，反之输出结果为空。模型结构如上图所示[12]。

# 5 实验结果

## 5.1 事件类型

识别事件类型结果如下，采用5折交叉验证，验证集结果：

|  |  |  |  |
| --- | --- | --- | --- |
| k-fold | Precision | Recall | F1 |
| 1 | 0.9459 | 0.9653 | 0.9555 |
| 2 | 0.9361 | 0.9575 | 0.9467 |
| 3 | 0.9477 | 0.9616 | 0.9546 |
| 4 | 0.9456 | 0.9579 | 0.9517 |
| 5 | 0.9457 | 0.9583 | 0.9519 |
| Average | 0.9442 | 0.9601 | 0.9521 |

使用全量训练集进行训练，测试集的结果如下：

|  |  |  |
| --- | --- | --- |
| Precision | Recall | F1 |
| 0.9459 | 0.9596 | 0.9527 |

## 5.2 论元抽取

下表中指标均为使用全量训练集进行训练，测试集的结果如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | Precision | Recall | F1 |
| Seq\_label | 0.6358 | 0.6711 | 0.6530 |
| SPN | 0.6449 | 0.6485 | 0.6467 |
| MRC\_se | **0.7356** | 0.7564 | **0.7459** |
| MRC\_io | 0.6940 | **0.7615** | 0.7262 |
| MRC\_readers | 0.7204 | 0.6917 | 0.7058 |

Seq\_label：序列标注模型；SPN：三元组关系抽取模型；MRC\_se：MRC 使用标注 span 开始和结束位置；MRC\_io：标注 span 全部字所在位置；MRC\_readers：使用略读和精读两阶段模型。

采取的评测指标是完全匹配，指标值较低。然后我们按照当时百度的字符级评测指标计算了一下，F1 值为83.5%。这个结果在1千多支队伍里排名前4%，当时第一名的结果是85.9%。

# 6 实验分析

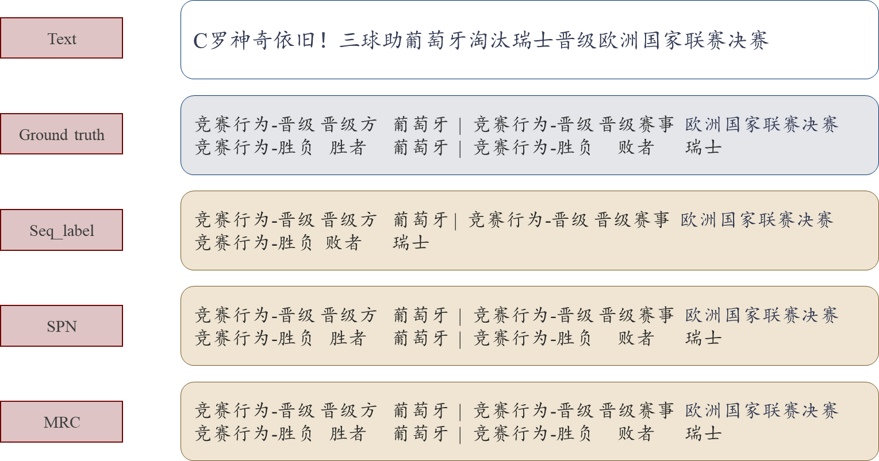
使用基于序列标注的模型，是 one-step 结构，同时识别事件类型和抽取论元角色对应的短语。没有 pipeline 结构带来的级联误差。但是序列标注方式每个字符只有一个标签，不能解决角色重叠和论元重叠的问题。

使用基于三元组关系抽取的模型，是 one-step 结构，没有级联误差。但是因为本实验数据集中无标准的 object，使用 trigger 作为替代。实验结果表明 precision 值高于序列标注，但 recall 值较低，综合 f1 值效果不如序列标注。效果不佳可能是因为一个 trigger 对应一个事件，所有属于同一事件类型的三元组的 object 均相同，导致模型不能准确区分不同的三元组。

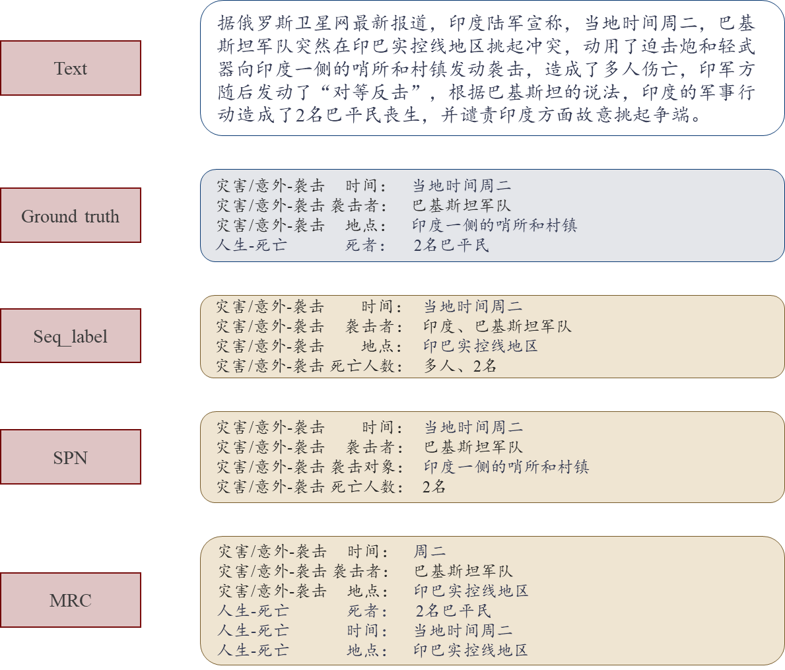
使用机器阅读理解的模型，是 two-step 结构，先识别事件类型，再抽取论元。不同事件类型和角色拼接得到组合标签，可解决多个事件类型识别和角色重叠问题。对每一个组合标签询问一次问题，可解决论元重叠问题。若一个句子包含多个论元角色，那么就将询问多次。此设计相当于扩充数据。但与此同时，增加了模型处理答案有无的误差。

对于 span 的不同标注方式，标注 span 开始和结束位置的 precision 值高于标注 span 全部字所在位置，但 recall 值低。标注 span 全部字所在位置 precision 值较低的原因是没有开始和结束位置的分界，模型会输出许多零碎的答案。而标注 span 开始和结束位置 recall 值较低的原因是一个句子里大多数字符不是论元短语，在标注 span 开始和结束位置时，负样本（即标注为0）数量远多于正样本。

MRC 略读和精读两阶段模型，其中略读阶段预测有无答案，准确率为88%。效果不佳可能是问题设计需要改进，样本标注不准确等。精读阶段有两个子任务：预测有无答案和输出 span 开始和结束位置，属于 joint model 结构，效果不如单独任务，说明此结构不能同时有效处理两个子任务，模型有待改进。



对于第一个例子，葡萄牙既是晋级事件的晋级方，也是胜负事件的胜者。序列标注只能输出晋级方这一个角色，而关系抽取和阅读理解模型都抽取出所有角色，解决角色重叠问题。



对于第二个例子，可以看到阅读理解方式另外两个，能抽取出更多信息。尽管这些信息不在真实的标签中，但是可以看出有一定的实际意义。

使用阅读理解模型输出的bad case 分析示例如下图：



# 7 实验过程总结与感想

马心睿：本次实验使用序列标注、三元组关系抽取和机器阅读理解三种方式识别事件类型和抽取论元。实验中发现自然语言处理是一个有趣的领域，尝试让擅长处理数字计算的机器来理解人类的思考方式，通过对自然语言编码和使用大规模语料预训练模型，使得机器在某些领域的表现超过人类平均表现，如SQuAD2.0[15]。在阅读理解领域，抽取式回答的效果已经很好，新的挑战是解决生成式回答，这需要让机器实现推理能力[16]。通过此次实验，从零到一学习了自然语言处理事件抽取领域相关知识，锻炼了使用实验验证想法的能力。

# 8 实验分工

# 参考文献

[1] Zhao, Yanyan, Bing Qin, Wan-xiang Che, and Ting Liu. "Research on Chinese event extraction." *Journal of Chinese Information Processing* 22, no. 1 (2008): 3-8.

[2] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

[3] Sun, Yu, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. "Ernie 2.0: A continual pre-training framework for language understanding." In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, pp. 8968-8975. 2020.

[4] Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. "Xlnet: Generalized autoregressive pretraining for language understanding." *arXiv preprint arXiv:1906.08237* (2019).

[5] Liu, Liyuan, Jingbo Shang, Xiang Ren, Frank Xu, Huan Gui, Jian Peng, and Jiawei Han. "Empower sequence labeling with task-aware neural language model." In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1. 2018.

[6] Qiu, Boyu, Xu Chen, Jungang Xu, and Yingfei Sun. "A survey on neural machine reading comprehension." *arXiv preprint arXiv:1906.03824* (2019).

[7] Li, Xiaoya, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. "A unified MRC framework for named entity recognition." *arXiv preprint arXiv:1910.11476* (2019).

[8] <https://aistudio.baidu.com/aistudio/projectdetail/545914>

[9] Segal, Elad, Avia Efrat, Mor Shoham, Amir Globerson, and Jonathan Berant. "A simple and effective model for answering multi-span questions." *arXiv preprint arXiv:1909.13375* (2019).

[10] Liu, Jian, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. "Event extraction as machine reading comprehension." In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1641-1651. 2020.

[11] Hu, Minghao, Furu Wei, Yuxing Peng, Zhen Huang, Nan Yang, and Dongsheng Li. "Read+ verify: Machine reading comprehension with unanswerable questions." In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 6529-6537. 2019.

[12] Zhang, Zhuosheng, Junjie Yang, and Hai Zhao. "Retrospective reader for machine reading comprehension." *arXiv preprint arXiv:2001.09694* (2020).

[14] Sui, Dianbo, Yubo Chen, Kang Liu, Jun Zhao, Xiangrong Zeng, and Shengping Liu. "Joint entity and relation extraction with set prediction networks." *arXiv preprint arXiv:2011.01675* (2020).

[15] <https://rajpurkar.github.io/SQuAD-explorer/>

[16] <https://allennlp.org/drop.html>