

Q1 Data processing :

1. Tokenizer

助教給的 train.json 檔如下圖:

```
{
  "id": "ab39567999fd376480ac3076904e598e",
  "question": "舍本 and 誰的數據能推算出連星的恆星的質量？",
  "paragraphs": [
    5234,
    6952,
    8264,
    836,
    92,
    2018
  ],
  "relevant": 836,
  "answers": [
    {
      "text": "斯特魯維",
      "start": 108
    }
  ]
},
```

我將所有的 paragraphs 接在一起變成一整大篇的文章，並且重新計算 answer start 的位置，也因為這樣做的關係我避免了做分類的任務(判斷 relevant)，當初會想這樣做是因為覺得如果把功課分成兩個任務去做，這樣整體準確率不就會變成 accuracy of stage 1 * accuracy of stage 2，這樣可能導致我的準確率較低(但結果並不是我想的那樣)，而雖然我們文章長度變為原本的 6~7 倍，但是因為 bert model 有使用 doc_stride 因此應該不會降低模型準確率太多，於是就這樣做下去了。

```
"paragraphs": [
  {
    "context": "所有的恒星，有生之年的绝大部分时间都是主序星，主要是燃烧氢元素，经由核融合产生能量。然而，不同质量的恒星在其演化阶段有着截然不同的性质。大质量恒星不仅寿命的命途和低质量恒星不同，它们的亮度和对周围环境的影响也不同。因此，天文学家经常以质量将恒星分成不同的群组。非常低质量的恒星是质量少于0.5太阳质量的恒星，不会演化进入红巨星分支，但是会直接成为白矮星。低质量恒星是质量超过0.5太阳质量，但未超过1.8-2.2太阳质量的恒星，会演化进入渐近巨星分支，在那里演化出复杂的核壳，中低质量恒星会经历多次融合和演化出复杂的核-壳层。大质量恒星的質量至少是7-10太阳質量，但也可能低至5或6太阳質量。这些恒星在生命的后期阶段最终融合，並以核心坍缩的超新星爆炸结束一生。心理学是否为自然科学的课题，目前也有争议。一般较宽泛接受的说法是心理学同时包含在自然科学與社會科學的範疇之中。自然科学的根本目的在于寻找隐藏在自然现象背后的规律，但是自然科学的工作尚不包括研究为什么存在这些规律。自然科学认为超自然的、随意的和自相矛盾的现象是不存在的。自然科学的最重要的两个支柱是观察和逻辑推理。由对自然的观察和逻辑推理自然科学可以引发出大自然中的规律。假如观察的现象与规律的预言不同，那么要么是观察中有错误，要么是因此至此为止被认为是正确的规律是错误的。一个超自然因素是不存在的。這是一個長久以來就存在的假設，大多數的恒星都是長期高在特定重力場的多星或聯星系統。特别是许多大质量的联星和道型恒星，有80%被认为是多星系统的一部分。然而，质量越低的恒星，星团存在的比例则越高，只有25%的红矮星被发现有伴星。因为95%的恒星是红矮星，所以在银河系内多数的恒星都是星团诞生的。恒星在宇宙中的分布是不均匀的，並且經常都是成群的與星團間的氣體、塵埃一起存在於星系中。一個典型的星系擁有數千億顆的恒星，而在可觀測宇宙中的星系數量超過一千億個。2018年對恒星數量的估計是在可觀測宇宙中有3000兆顆。儘管人們往往認為恒星僅存在於星系中，但星際的恒星已經被發現了。在19世紀雙星觀測所獲得的成就重要性也增加了。在1834年，白塞爾觀測到天狼星自行運動的變化，因而推測有一顆隱蔽的伴星；愛德華·皮克林在1899年觀測到隨週期性分裂的光譜線時發現第一顆白矮星，週期是184天。天文學家斯特魯維和喬本·波德爾，但於細仔細的觀察和收集了許多聯星的資料，使得可以從被確定的軌道要素推算出伴星的質量。第一個獲得解答的是1827年由赫利克斯·赫利星透過望遠鏡的觀測得到的聯星軌道。對恒星的科學研究在20世紀變得快速的進展，相片成為天文學上有價值的工具。卡爾·史瓦西發現經由比較視星等和攝影星等的差別，可以得到恒星的顏色和它的溫度。1921年，光電光度計的發展可以在不同的波長間區上非常精密的測量等等。阿爾伯·特·溫克生在虎克望遠鏡第一次使用干涉儀測量出恒星的直徑。超新星爆炸會把這顆恒星的大部分物質都吹散出去。剩下的就是中子星，或是在質量比大恒星就會形成黑洞。在中子星內的物質是中子簡併物質，和一種可能存在于核心但極不穩定的簡併物質，夸克物質。物質在黑洞中心所處的狀態是迄今仍不了解的。死後恒星拋出殼的外層物質包括一些重元素，可能恒星形成的世代交替中成為新恒星的原料。這些重元素可以形成碳星的行星。從超新星和大恒星的恒星風拋出的物質在星際物質的構成中扮演著重要的角色。除了星團的恒星之外，恒星系統可以是兩顆或更多的恒星受到重力的約束而在軌道上互繞著。最普遍的恒星系統就是聯星，但是也發現有三顆或更多恒星的系統。而因為軌道要穩定的緣故，這些恒星系統，經常會形成階梯度的共軌恒星。也存在著更大的、被稱為星團的集團。他們的範圍從只有幾十顆恒星，鬆散的星團，到龐大的擁有數十萬顆恒星，稱為球狀星團的集團。1930年，印度物理學家蘇布拉馬尼揚·錢德拉塞卡根據廣義相對論計算，出質量大於1.4倍太陽質量的非轉動星體會因重力塌縮成為電子簡併態。愛丁頓雖然在理論上支持黑洞存在的可能性，但同時認為錢德拉塞卡的觀點事實上不能成立，他認為「應當有某種自然定律阻止恒星出現這種荒謬的行為」。當時的物理學家如盧瑟、亨利·摩塞等人也贊同錢德拉塞卡的理論，但出於愛丁頓質疑的原因，他們也沒有公開對錢德拉塞卡表示支持。不過從某種意義上說，愛丁頓也是正確的，當恒星質量大於錢德拉塞卡極限後，確實仍然會中子簡併壓力阻止恒星繼續塌縮。到了1939年，美國物理學家羅伯特·歐本海默等人推算了這種情形的恒星質量上限，這個極限被稱作托爾曼-歐本海默-沃爾科夫極限。當今的天體物理學家普遍認為，除非有如未知的夸克簡併壓力一樣因素的存在，質量大於托爾曼-歐本海默-沃爾科夫極限的恒星將最終會塌縮為錢德拉塞卡所預言的黑洞。即使如此，史瓦西解作為當時能夠描述黑洞行為的唯一一解，由於具有一些讓人看似不大完美的性質以及難以與實驗觀測相聯繫，一直沒有進入主流物理學研究的視野。關於黑洞的理論乃至整個廣義相對論領域的研究由此擴充了二十年之久。",
    "qas": [
      {
        "question": "舍本 and 誰的數據能推算出連星的恒星的質量？",
        "id": "ab39567999fd376480ac3076904e598e",
        "answers": [
          {
            "text": "斯特魯維",
            "answer_start": 977
          }
        ]
      }
    ]
  }
]
```

而在丟資料進 bert model 的方式如下圖顯示

tokens: [CLS] 平安京的設計是採用中國什麼時代的京都格局？[SEP] 斯就已經有關於音樂的研究，古希臘的阿夫洛斯管及里拉琴也漸漸發展為

也就是 [CLS] question [SEP] paragraph context [SEP]，每個字會去 bert 的 vocab.txt 檔(如下圖) 找尋對應的 id，[CLS] 起頭符號、[SEP] 分割符號以及 [UNK] 未知詞。

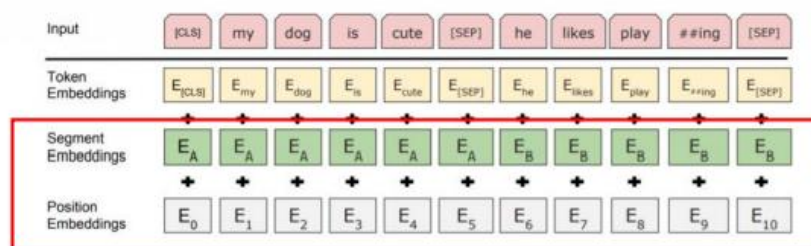
975	偏
976	偕
977	倣
978	倣
979	倣
980	倣
981	倣
982	倣
983	倣
984	倣
985	倣
986	倣
987	倣
988	倣
989	倣
990	倣
991	倣

由於使用 `doc_stride` 的關係，且 `bert input size` 最大只有 512 (我最大設 200 個字，因為要在 8G 的空間上 `train RoBERTa large`)故需要把文章拆成許多小文章，這樣就必須要做(token to original map)的動作，`input id` 為把 `input` 每個字對應到 `vocab.txt` 的索引，`segment ids` 就是把 `question` 用 0 表示，`paragraph context` 用 1 表示，下圖可見：

```
token_to_orig_map: 23:1597 24:1598 25:1599 26:1600 27:1601 28:1602 29:1603 30:1604 31:1605 32:1606 33:1607 34:1608 35:1609 36:1610 37:1611 38:1612 39:1613 40:1614 41:1615 42:1616 43:1617 44:1618 45:1619 46:1620 47:1621 48:1622 49:1623 50:1624 51:1625 52:1626 53:1627 54:1628 55:1629 56:1630 57:1631 58:1632 59:1633 60:1634 61:1635 62:1636 63:1637 64:1638 65:1639 66:1640 67:1641 68:1642 69:1643 70:1644 71:1645 72:1646 73:1647 74:1648 75:1649 76:1650 77:1651 78:1652 79:1653 80:1654 81:1655 82:1656 83:1657 84:1658 85:1659 86:1660 87:1661 88:1662 89:1663 90:1664 91:1665 92:1666 93:1667 94:1668 95:1669 96:1670 97:1671 98:1672 99:1673 100:1674 101:1675 102:1676 103:1677 104:1678 105:1679 106:1680 107:1681 108:1682 109:1683 110:1684 111:1685 112:1686 113:1687 114:1688 115:1689 116:1690 117:1691 118:1692 119:1693 120:1694 121:1695 122:1696 123:1697 124:1698 125:1699 126:1700 127:1701 128:1702 129:1703 130:1704 131:1705 132:1706 133:1707 134:1708 135:1709 136:1710 137:1711 138:1712 139:1713 140:1714 141:1715 142:1716 143:1717 144:1718 145:1719 146:1720 147:1721 148:1722 149:1723 150:1724 151:1725 152:1726 153:1727 154:1728 155:1729 156:1730 157:1731 158:1732 159:1733 160:1734 161:1735 162:1736 163:1737 164:1738 165:1739 166:1740 167:1741 168:1742 169:1743 170:1744 171:1745 172:1746 173:1747 174:1748 175:1749 176:1750 177:1751 178:1752 179:1753 180:1754 181:1755 182:1756 183:1757 184:1758 185:1759 186:1760 187:1761 188:1762 189:1763 190:1764 191:1765 192:1766 193:1767 194:1768 195:1769 196:1770 197:1771 198:1772 199:1773 200:1774 201:1775 202:1776 203:1777 204:1778 205:1779 206:1780 207:1781 208:1782 209:1783 210:1784 211:1785 212:1786 213:1787 214:1788 215:1789 216:1790 217:1791 218:1792 219:1793 220:1794 221:1795 222:1796 223:1797 224:1798 225:1799 226:1800 227:1801 228:1802 229:1803 230:1804 231:1805 232:1806 233:1807 234:1808 235:1809 236:1810 237:1811 238:1812 239:1813 240:1814 241:1815 242:1816 243:1817 244:1818 245:1819 246:1820 247:1821 248:1822 249:1823 250:1824 251:1825 252:1826 253:1827 254:1828 255:1829 256:1830 257:1831 258:1832 259:1833 260:1834 261:1835 262:1836 263:1837 264:1838 265:1839 266:1840 267:1841 268:1842 269:1843 270:1844 271:1845 272:1846 273:1847 274:1848 275:1849 276:1850 277:1851 278:1852 279:1853 280:1854 281:1855 282:1856 283:1857 284:1858 285:1859 286:1860 287:1861 288:1862 289:1863 290:1864 291:1865 292:1866 293:1867 294:1868 295:1869 296:1870 297:1871 298:1872 299:1873 300:1874 301:1875 302:1876 303:1877 304:1878 305:1879 306:1880 307:1881 308:1882 309:1883 310:1884 311:1885 312:1886 313:1887 314:1888 315:1889 316:1890 317:1891 318:1892 319:1893 320:1894 321:1895 322:1896 323:1897 324:1898 325:1899 326:1900 327:1901 328:1902 329:1903 330:1904 331:1905 332:1906 333:1907 334:1908 335:1909 336:1910 337:1911 338:1912 339:1913 340:1914 341:1915 342:1916 343:1917 344:1918 345:1919 346:1920 347:1921 348:1922 349:1923 350:1924 351:1925 352:1926 353:1927 354:1928 355:1929 356:1930 357:1931 358:1932 359:1933 360:1934 361:1935 362:1936 363:1937 364:1938 365:1939 366:1940 367:1941 368:1942 369:1943 370:1944 371:1945 372:1946 373:1947 374:1948 375:1949 376:1950 377:1951 378:1952 379:1953 380:1954 381:1955 382:1956 383:1957 384:1958 385:1959 386:1960 387:1961 388:1962 389:1963 390:1964 391:1965 392:1966 393:1967 394:1968 395:1969 396:1970 397:1971 398:1972 399:1973 400:1974 401:1975 402:1976 403:1977 404:1978 405:1979 406:1980 407:1981 408:1982 409:1983 410:1984 411:1985 412:1986 413:1987 414:1988 415:1989 416:1990 417:1991 418:1992 419:1993 420:1994 421:1995 422:1996 423:1997 424:1998 425:1999 426:2000 427:2001 428:2002 429:2003 430:2004 431:2005 432:2006 433:2007 434:2008 435:2009 436:2010 437:2011 438:2012 439:2013 440:2014 441:2015 442:2016 443:2017 444:2018 445:2019 446:2020 447:2021 448:2022 449:2023 450:2024 451:2025 452:2026 453:2027 454:2028 455:2029 456:2030 457:2031 458:2032 459:2033 460:2034 461:2035 462:2036 463:2037 464:2038 465:2039 466:2040 467:2041 468:2042 469:2043 470:2044 471:2045 472:2046 473:2047 474:2048 475:2049 476:2050 477:2051 478:2052 479:2053 480:2054 481:2055 482:2056 483:2057 484:2058 485:205
```

而標準的 bert input 有三層:

Token Embeddings 、 Segment Embeddings 、 Position Embeddings



2. Answer Span

(a) How did you convert the answer span start/end position on characters to position on tokens after BERT tokenization?

- c. the loss function you used.

Loss 分成 start logit 以及 end logit 兩個部分:

```
loss_fct = CrossEntropyLoss()
start_loss = loss_fct(start_logits, start_positions)
end_loss = loss_fct(end_logits, end_positions)
total_loss = (start_loss + end_loss) / 2
```

- d. The optimization algorithm (e.g. Adam), learning rate and batch size.

Implements BERT version of Adam algorithm with weight decay fix

BertAdam: lr=3e-5

warmup_proportion=0.1(前面 10 萬的 data 會使用較小的 LR 訓練)

b1=0.9, b2=0.98, e=1e-6, weight_decay=0.01

2. Try another type of pretrained model and describe (2%)

- a. your model

Chinese-bert-base

- b. performance of your model.

EM: 68

F1: 71

- c. the difference between pretrained model

RoBERTa 採用的是 dynamic masking 方式，且在 optimizer adam 超參數 β_2 改為 0.98 這樣一來使的 training 在大的 batch size 比較穩定，在訓練數據上讓 model 看了更多的資料，包含了 BookCorpus, CC-News, OpenWebText, Stories，這也是 RoBERTa 比 Bert 還要強大的主要原因。

在 model config 上得差異如下:

Chinese-bert-base:

```
{
  "attention_probs_dropout_prob": 0.1,
  "directionality": "bidi",
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "max_position_embeddings": 512,
  "num_attention_heads": 12,
  "num_hidden_layers": 12,
  "pooler_fc_size": 768,
  "pooler_num_attention_heads": 12,
  "pooler_num_fc_layers": 3,
  "pooler_size_per_head": 128,
  "pooler_type": "first_token_transform",
  "type_vocab_size": 2,
  "vocab_size": 21128
}
```

Chinese-roberta-wwm-ext-large:

```
{
  "architectures": [
    "BertForMaskedLM"
  ],
  "attention_probs_dropout_prob": 0.1,
  "bos_token_id": 0,
  "directionality": "bidi",
  "eos_token_id": 2,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 1024,
  "initializer_range": 0.02,
  "intermediate_size": 4096,
  "layer_norm_eps": 1e-12,
  "max_position_embeddings": 512,
  "model_type": "bert",
  "num_attention_heads": 16,
  "num_hidden_layers": 24,
  "output_past": true,
  "pad_token_id": 1,
  "pooler_fc_size": 768,
  "pooler_num_attention_heads": 12,
  "pooler_num_fc_layers": 3,
  "pooler_size_per_head": 128,
  "pooler_type": "first_token_transform",
  "type_vocab_size": 2,
  "vocab_size": 21128
}
```

- d. The optimization algorithm (e.g. Adam), learning rate and batch size.

Implements BERT version of Adam algorithm with weight decay fix

BertAdam: lr=3e-5

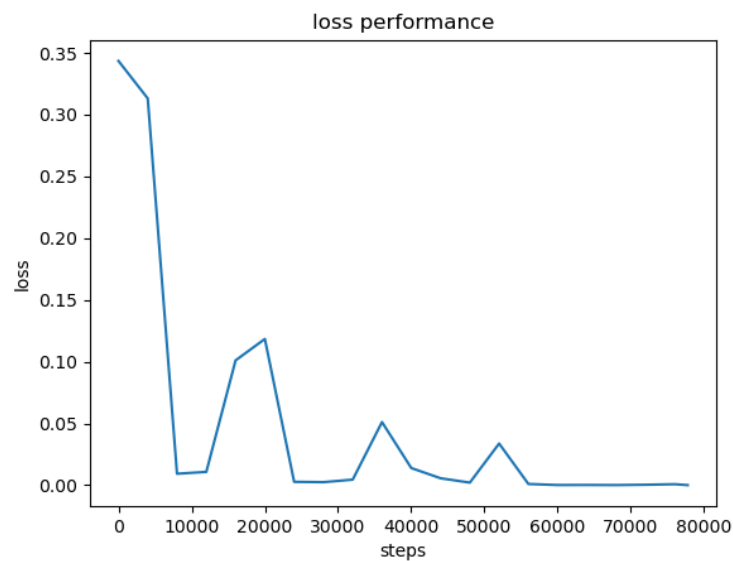
warmup_proportion=0.1(前面 10 萬的 data 會使用較小的 LR 訓練)

b1=0.9, b2=0.999, e=1e-6, weight_decay=0.01

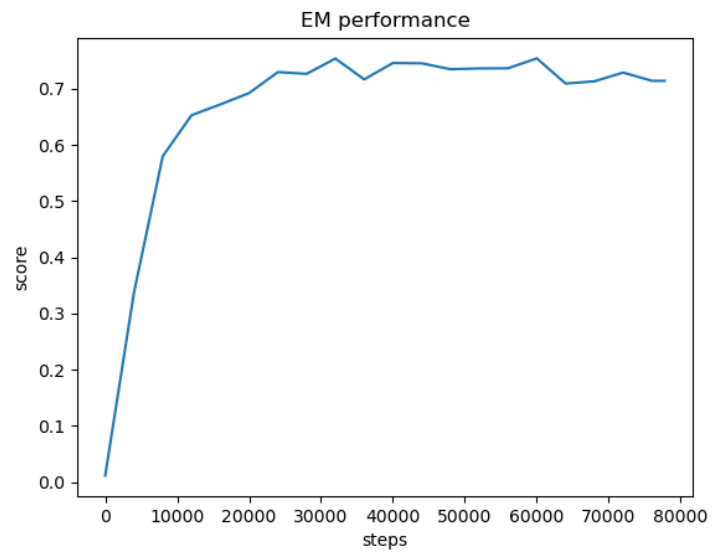
Q3: Curves

使用 matplotlib 套件繪出，每訓練 4000 步得到一組資料，且為第一個 epoch。

1. Plot



a. learning curve of EM



b. learning curve of F1

