DeepL
订阅DeepL Pro以翻译大型文件。
欲了解更多信息，请访问www.DeepL.com/pro。

COMP90042项目2023：
气候科学声明的自动事实核查

墨尔本大学版权所有，2023年

项目类型：个人
报告和代码提交截止日期：2023年5月15日星期一晚上9点
Codalab提交截止日期：2023年5月15日（星期一）下午1点**（此部分不能延期）**。

气候变化对人类的影响是一个重大问题。然而，有关气候科学的未经证实的声明的增加导致了公众舆论的扭曲，突出了对与气候科学有关的声明进行事实检查的重要性。请考虑以下说法和相关证据：

声称：地球的气候敏感性很低，大气中的二氧化碳增加一倍将导致地表温度变化在1°C左右或更少。

证据：

1. 在他关于这个问题的第一篇论文中，他估计，如果CO 2的数量增加一倍，全球温度将上升约5至6℃（9.0至10.8°F）。
2. 1990年IPCC第一次评估报告估计，平衡气候对二氧化碳增加的敏感性在1.5至4.5℃（2.7至8.1°F）之间，"根据现有知识的最佳猜测 "为2.5℃（4.5°F）。

应该不难看出，这个说法没有证据段落的支持，而且假设证据来源是可靠的，这样的说法就是误导。该项目面临的挑战是开发一个自动事实核查系统，在该系统中，给定一个主张，目标是从知识来源中找到相关的证据段落，并对该主张是否得到证据支持进行分类。

更具体地说，你将得到一个索赔清单和一个包含大量证据段落的语料库（"知识源"），而你的系统必须：(1)从知识源中搜索出与索赔最相关的证据段落；(2)根据证据将索赔的状态分为以下4类：{支持、反驳、信息不充分、有争议}。为了建立一个成功的系统，它必须能够检索到正确的证据段落集，*并*对索赔进行正确分类。

除了系统实施，你还必须写一份报告，描述你的事实核查系统，例如，检索和分类组件如何工作，你所做的选择背后的原因以及系统的性能。我们希望你会喜欢这个项目。为了使它更有吸引力，**我们将以Codalab竞赛的方式进行任务**。你将与班上的其他学生进行竞争。下面几节将详细介绍数据格式、系统评估、评分标准和Codalab的使用。你的评估将根据你的报告、你在比赛中的表现和你的代码来打分。

提交材料： 请提交以下材料：

- 报告（.pdf）：https://canvas.lms.unimelb.edu.au/courses/151109/assignments/387847

- 如果使用Unix工具，一个包含你的Python代码（.py或.ipynb）和脚本代码（.sh或类似）的压缩文件（.zip）： https://canvas.lms.unimelb.edu.au/courses/151109/assignments/387859

请注意，项目报告和代码有两个不同的提交链接/外壳。之所以将这两个提交文件分开，是因为我们将在项目完成后不久对报告进行同行评审。请注意，你应该为你的报告上传一个pdf文件，为你的代码上传一个zip文件；**所有其他格式都不允许**，例如docx、7z、rar等。如果你使用这些其他格式，你的提交将不会被标记，并将被打0分。你不需要在压缩包中上传你的数据文件（例如，证据段落）。你会在提交外壳中找到更多关于如何提交报告和代码的信息。

如果包含多个代码文件，请在每个文件的标题中明确说明其作用。如果使用了预训练的模型或嵌入，你不需要把它们作为提交的一部分，但请确保你的代码或脚本在必要时下载它们。请注意，如果需要，我们可能会审查你的代码，但是请注意，代码是次要的--评分的主要重点将是你的报告和你的系统在Codalab上的表现。

你必须向Codalab竞赛提交至少一份作品。

逾期提交：每天-10%。

分数：科目的35%。

材料：关于COMP90042所需的基本设置，包括iPython笔记本浏览器和Python软件包NLTK、 Numpy、Scipy、 Matplotlib、 Scikit-Learn和Gensim，请参见Canvas上的使用Jupyter笔记本和Python页面（模块>资源下）。对于这个项目，我们鼓励你使用NLTK提供的NLP工具，如斯坦福分析器、 NER标签器等，或者你可以选择使用Spacy或AllenNLP工具包，它们捆绑了很多优秀的NLP工具。你也可以使用基于Python的深度学习库：TensorFlow、 Keras或PyTorch。**你应该使用Python 3.8。**

对于这个项目，你可以使用预训练的语言模型或词嵌入。然而，请注意，你**不允许**：(1)使用闭源模型，例如OpenAI GPT-3；(2)不能在Colab上运行的模型（例如在Colab的GPU上不适合的非常大的模型）；或者(3)以任何形式寻找更多的训练数据（无论它们是相关或不相关任务的标记或未标记）来训练你的系统。换句话说，你应该只使用所提供的数据来训练你的系统，这些数据包括一个训练集、一个开发集和一个测试集；关于它们的格式和用法，请看下面 "数据集 "的说明。如果你想使用一些东西，但你不确定它是否被允许，请在讨论区询问（不要把你的想法过多地泄露给全班同学）。

分数：你的评分将基于几个标准：报告表达的清晰度、方法的合理性和新颖性、工作的实质、结果的解释和系统的性能（下文 "评分 "部分将提供更多细节）。

更新：项目的任何重大变化都将通过Canvas公布。小的改动和澄清将在Canvas上的讨论区公布；我们建议你定期查看。

学术不端行为：虽然你可以自由地与其他学生讨论项目，但学生之间重复使用代码，从网上复制大块的代码，或其他明显影响的情况，将被视为作弊。请记得正确引用你的来源，包括研究思路和算法解决方案以及代码片段。我们将检查提交的材料是否具有原创性，如果认为发生了不适当的串通或剽窃行为，将援引大学的学术不端行为政策。关于在学术诚信方面使用人工智能工具的问题，请参见大学的相关声明：https://academicintegrity.unimelb.edu. au/plagiarism-and-collusion/artificial-intelligence-tools-and-technologies。

数据集

为你提供了几个项目的文件：

[train-claims,dev-claims].json：标记的训练和开发集的JSON文件；

[test-claims-unlabelled].json：未标记的测试集的JSON文件；
evidence.json：包含大量证据段落的JSON文件（即 "知识源"）； dev-claims-baseline.json：JSON文件，包含基线系统对开发集的预测； eval.py：用于评估系统性能的Python脚本（详见下文 "评估"）。

对于标记的索赔文件（train-claims.json, dev-claims.json），每个实例都包含索赔ID、索赔文本、索赔标签（四个类别之一：{SUPPORTS, REFUTES, NOT_ENOUGH_INFO, DISPUTED}）和证据ID列表。未标记的索赔文件（test-claims-unlabelled.json）具有类似的结构，只是它只包含索赔ID和索赔文本。更具体地说，标记的索赔文件有以下格式：

```
{
  "索赔-2967"：
  {
    claim_text："[南澳大利亚]拥有世界上最昂贵的电力。" claim_label："支持"
    证据：["evidence-67732", "evidence-572512"]
  },
  "索赔-375"：
  ...
}
```

证据ID的列表（如evidence-67732，evidence-572512）来自于《中国》的证据段落。

证据.json：

```
{
  "证据-0"："约翰-本尼特-劳斯，英国企业家和农业科学家"，"证据-1"："林德伯格在
  16岁时开始其职业生涯，最终....."、
  ...
}
```

给定一个索赔（例如索赔-2967），你的系统需要从evidence.json中搜索和检索最相关的证据段落列表，并对索赔进行分类（上述4类中的1类）。你应该至少检索出一个证据段落。

训练集（train-claims.json）应被用于建立你的模型，例如用于开发特征、规则和启发式方法，以及用于监督/非监督学习。我们鼓励你仔细检查这些数据，以充分理解这项任务。

开发集（dev-claims.json）的格式与训练集一样。这将帮助你做出主要的实施决策（例如，选择最佳的超参数配置），而且还应该用于对你的系统进行详细的分析--包括测量性能和错误分析--在报告中。

你将使用测试集（test-claims-unlabelled.json）来参加Codalab比赛。由于这个原因，我们没有为这个分区提供标签（即证据段落和索赔标签）。我们允许（并鼓励）你在训练集和开发集上训练你的最终系统，以便在测试集上最大限度地提高性能，但你**在任何时候都**不应该手动检查测试数据集；任何表明你已经这样做的迹象都会导致失分。就系统输出的格式而言，我们为此提供了dev-claims-predictions.json。注意：你会注意到它的格式与标记的索赔文件（train-claims.json或dev-claims.json）相同，尽管claim_text字段是可选的（即我们在评估时不使用这个字段），你可以自由地省略它。

评价

我们提供一个脚本（eval.py）来评估你的系统。这个脚本需要两个输入文件，即地面真相和你的预测，并计算三个指标：（1）证据检索的F分数；（2）索赔分类的准确性；以及（3）证据检索的F分数和索赔分类准确性的谐波平均值。下面显示的是一个基线系统在开发集上运行预测的结果：

```
$ python eval.py --predictions dev-claims-baseline.json --groundtruth dev-claims.json 证据检
索 F-score (F)                    = 0.3377705627705628
```

声称的分类精度（A）=0.35064935064935066 F和A的谐
波平均值 = 0.3440894901357093

这三个指标的计算方法如下：

1. 证据检索F-score（F）：计算系统检索的证据段落与地面真相证据段落的匹配程度。对于每一项索赔，我们的评估考虑了*所有*检索到的证据段落，通过与地面真相的比较，计算出精度、召回率和F分。

语录，并通过对所有索赔的平均数来汇总F分数。例如，给定一个主张，如果系统检索到以下集合{证据-1，证据-2，证据-3，证据-4，证据-5}，而基础真理集是{证据-1，证据-5，证据-10}，那么精度=2/5，召回=2/3，F分数=1/2。该
这个指标的目的是衡量你的事实核查系统的检索部分工作得如何。

2. 索赔分类精度（A）：计算索赔标签预测的标准分类精度，忽略系统检索的证据段落集。这个指标只评估系统对索赔的分类情况，旨在了解你的事实核查系统的分类部分的工作情况。

3. F和A的谐波平均值：计算证据检索的F分数和索赔分类的准确性的谐波平均值。请注意，这个指标是在我们获得总的（所有索赔的）F分数和准确率后，在最后计算的。这个指标旨在评估系统的检索和分类部分，因此将被**用作Codalab上系统排名的主要指标**。

前两个指标（F-score和准确性）是用来帮助诊断和发展你的系统的。虽然它们不用于在排行榜上对你的系统进行排名，但你应该在报告中记录它们，并利用它们来讨论你的系统的优势/劣势。

示例预测文件，dev-claims-baseline.json，是开发集上的基线系统的输出。这个文件将帮助你了解创建你的开发输出（用于使用eval.py调整你的系统）和测试输出（用于提交给Codalab比赛）所需的文件格式。

请注意，这不是一个*现实的*基线，你可能会发现你的系统表现比它差。原因是这个基线以下列方式构建其输出：（1）索赔标签是随机选择的；（2）证据段落集结合了几个随机选择的地面真相段落和几个随机选择的知识源段落。我们创造了这样一个 "基线"，因为一个*真正的随机基线*，即随机选择一组证据段落，很可能会产生一个零的证据检索F-score（因此F-score和准确性的谐波平均值也是零），它不会作为一个好的诊断例子来解释指标。澄清一下，这个基线不会以任何方式用于对Codalab上提交的系统进行排名，而只是为了说明指标和一个系统输出的例子。

分级

你提交的材料将被打分，具体如下：

组成部分	标准	内容	分数
写作	清晰度表/图	报告是否写得很好，结构合理？	5
		表格和数字是否可以解释和有效使用？	3
内容	健全性	实验是否合理？方法是否合理，使用是否正确？	7
	价值观	做了多少工作？是否有足够的物质？	5
	新颖性	技术或方法的新颖性或雄心有多大？	5
	结果	结果和发现是否令人信服？它们是否得到了很好的阐述？	5
绩效	H.F和A的平均值	根据Codalab排行榜上的排名进行评分	5

应提交一份报告，对所使用的方法进行描述、分析和比较评估。你应该足够详细地描述你的方法，以便我们可以在不看你的代码的情况下复制它们。你应该提到你在实现你的系统时所作的任何选择，以及使用开发集对这些选择进行的经验论证。你还应该详细说明你的开发性能和Codalab排行榜上的 "最

终评价 "性能（详情见下面的章节）。你应该使用表格和适当的图表来报告你的结果/发现。

对你的方法的描述应该是清晰和简明的。你应该把它写成一个硕士生可以毫无困难地阅读和理解的水平。如果你使用任何现有的算法，你不必重写完整的描述，但必须提供一个显示你的理解的摘要，你应该提供相关文献中的参考文献的引用。在报告中，我们将非常希望看到证据

你选择一种方法而不是另一种方法的思维过程和理由（如 "合理性 "标准的较高权重所显示的）。

报告应以PDF格式提交，内容不超过4页A4纸，**不包括参考文献**。**不允许有附录**--因此，你应该仔细考虑你要在报告中包含的信息，以建立一个*连贯*而*简洁的*叙述。

你在写报告时应使用ACL模板。我们希望你使用LATEX，但你也可以使用Word。**你必须在标题下写上你的学生号**（使用LATEX中的author字段并启用aclfinalcopy选项），**但不要写上你的名字**，以便于匿名的同行评审。我们不接受超过上述限制的报告，或其他违反风格要求的报告。

对于性能部分，你将根据你的系统的相对排名进行评分，计算公式为：：$\frac{N-r+1}{N} \times 5$，其中N是排行榜上的系统总数，r是你的系统排名。例如，如果N=100，你是排名第一的系统（#1），你将得到5.0分；但如果你排名第50，你将得到2.55分。

鳕鱼

你需要加入Codalab上的比赛，提交你的事实核查系统。Codalab竞赛链接将在晚些时候在Canvas上公布。

你必须使用你的**student.unimelb.edu.au**地址的电子邮件来参加比赛（通过 "参与 "标签），你不允许用多个电子邮件账户参加比赛。**任何学生如果被发现用多个账户参加比赛，将被自动暂停比赛，并且该项目评分为零**。

一旦你参加了比赛，请点击右上角的登录名并选择 "设置 "来编辑你的账户信息。**用你的学生号设置你的队名。没有队名的参赛作品将不会被标记。**

要提交你的测试输出，选择 "参与 "标签，点击 "持续评估 "按钮，然后点击 "提交"。这将允许你选择一个文件，该文件被上传到Codalab服务器，它将评估你的结果并在排行榜上增加一个条目。你的文件应该是一个**压缩档案**，包含一个名为test-claims-predictions.json的文件。该JSON文件应该为test-claims-unlabelled.json中的所有索赔产生索赔标签和证据段落。JSON文件的格式应遵循提供的基线系统的格式（即dev-claims-baseline.json）。**如果文件名不同，系统将产生错误信息，因为它不能处理你的文件**。

结果显示在排行榜的 "结果 "标签下的 "正在进行的评估"。比赛于5月15日下午1点结束，此后将不再接受提交的作品（该期限不能延期）。此时，"最终评价 "结果将被公布。这两组结果反映了对测试数据的不同子集的评估。正在进行的评价中的最佳成绩可能不是最终评价中的最佳成绩，我们将在评估中使用最终评价的分数。**现在可以在报告中讨论你提交的最佳结果**，报告应在同一天（5月15日）晚上9点提交。

请注意，Codalab每天只允许每个用户提交3次，所以请只在你对你的系统做了有意义的改变后再上传你的结果。**请不要根据正在进行的测试集对你的系统进行过度调整**，因为当它在最终的测试集上进行评估时，很可能会出现性能下降，因为它很可能在正在进行的测试集上进行了过度拟合（我们每年都在COMP90042项目中看到这种情况，在持续评估期间有大量提交的系统在最终评估结果发布后排名会大幅下降）。请注意，Codalab的反应有时有点慢，所以你需要给它一分钟左右的时间来处理你的文件并更新结果表。