# NLP课程作业:临床术语标准化

## 背景与方法调研

临床术语标准化是医学统计中不可或缺的技术之一。在医疗领域，同一种诊断、手术、药物、检查、化验和症状通常以多种不同方式进行描述。

标准化的主要任务是将这些多样的表达方式映射到统一的标准术语。通过建立术语标准化的基础，研究人员能够进行进一步的电子病历分析。

**主要目标：**

针对中文电子病历中挖掘出的真实诊断实体，并以《国际疾病分类 ICD-10 北京临床版v601》词表为标准进行语义标准化。在给定一诊断原词，要求给出其对应的诊断标准词。

**主要内容：**

本次项目开发旨在探索并提高自然语言处理（NLP）在临床医学领域的应用。通过深入研究，实现临床术语标准化的自动化处理，包括实体标准化、关系抽取和概念归一化等任务。 该课题的重要性在于加速医学信息处理、医学研究和临床决策的创新。通过自动化术语标准化，可以提高医疗数据的质量和可比性，促进全球范围内医学数据的共享和合作。

研究发展历程始于对临床文本处理的深入研究。起初，研究侧重于了解医学领域的特殊术语和临床文本的语言结构。随后，随着深度学习方法在NLP领域的兴起，研究逐渐引入了这些先进技术，如循环神经网络（RNN）和长短时记忆网络（LSTM），以更好地捕捉文本中的序列信息。

随着预训练模型的成功应用，研究逐渐演变为使用BERT（Bidirectional Encoder Representations from Transformers）等模型，以在大规模数据上进行预训练，学习到更为丰富的语言表示。这一阶段强调了在自然语言处理中迁移学习的重要性，通过在通用任务上进行预训练，提高了系统在临床文本处理任务中的性能。

研究进一步关注了临床术语标准化任务的多模态处理，结合了图像和文本信息，以更全面地理解患者的医学记录。这个发展阶段着眼于处理来自不同源的医学数据，使得系统更具广泛适应性和实用性。

未来的研究方向可能包括对自监督学习的进一步探索，以减少对大规模标注数据的依赖。研究者还可能关注解释性模型的开发，以提高系统对医学决策的可解释性。

整个研究历程展示了NLP技术在临床医学中的快速发展，从基础的术语理解到深度学习和预训练模型的引入，再到多模态处理的探索。这一历程促使了医学信息技术领域的不断创新，为医疗决策提供更准确、全面的信息支持。

这一任务要求项目人员具备深厚的NLP和医学领域知识，能够处理大规模临床文本，并关注伦理和法律问题。该课题有望推动医学信息技术的发展，为医疗决策提供更准确、全面的信息支持。

本次课程项目开发仅要求对临床术语实现标准化，并对测试集进行完善，最终计算出模型与算法组合在课题上的F1指数。

## 方法或算法间比较与分析

1. **基于词嵌入的文本特征提取方法间的比较：**

**Word2vec**

Word2vec是名为Gensim的Python库中的算法之一。它可以通过调用gensim.models.Word2Vec来使用。Word2Vec将单词映射到连续的向量空间中，以捕捉单词之间的语义关系。以下是Word2Vec的一些优点和缺点：

**优点：**

* 易于使用： Gensim提供了一个简单的API，使得Word2Vec的实现相对容易。用户可以轻松地训练Word2Vec模型。
* 效果好： Word2Vec的词向量通常能够捕捉到单词之间的语义关系，例如单词之间的相似性和关联性。这对于自然语言处理任务，如文本分类、情感分析和信息检索等，非常有用。
* 可扩展性： Gensim支持在大型文本语料库上训练Word2Vec模型，因此它在处理大量文本数据时表现出色。
* 节省内存： Gensim的实现采用了内存高效的方法，可以有效地处理大规模语料库而不需要大量内存。

缺点：

* 调参需求： Word2Vec模型有一些超参数需要调整，如向量维度(dim)、窗口大小(window)、迭代次数(epochs)等。不同的任务和数据集可能需要不同的超参数设置，因此可能需要一些实验来获得最佳性能。
* 有监督学习的限制： Word2Vec是一种无监督学习方法，它只能生成词向量，而不能直接用于解决具体的监督学习任务。通常需要额外的工作来将Word2Vec嵌入应用到特定的NLP任务中。
* 数据依赖性： Word2Vec模型的性能高度依赖于训练数据的质量和数量。如果数据质量差或者过于有限，模型可能无法产生有用的词向量。

**BERT**

BERT（Bidirectional Encoder Representations from Transformers）是一种深度学习模型，广泛应用于自然语言处理领域。它的实现在许多深度学习框架中都可找到，其中包括Hugging Face的Transformers库。使用BERT可以通过简单地调用相应的API来实现。

**优点：**

BERT的优点之一是其在捕捉语义关系方面的卓越效果。与传统的单向语言模型不同，BERT采用双向编码器，能够同时考虑一个词的上下文信息，更好地理解单词之间的关联性。这种能力使得BERT在各种自然语言处理任务中表现出色，包括文本分类、命名实体识别、问答等。

与此同时，BERT模型也具备良好的可扩展性。它可以在大规模文本语料库上进行训练，适应不同领域和任务的需求。这种灵活性使得BERT在处理大量文本数据时表现优异。

**缺点：**

BERT模型的训练过程需要大量的计算资源和时间，对硬件要求较高。此外，调整BERT的超参数也是一个挑战，包括学习率、批量大小等参数的选择，需要进行一定的实验以获取最佳性能。最后，类似于Word2Vec模型，BERT也对训练数据的质量和数量敏感，差或有限的数据可能影响其性能。

综合而言，BERT模型以其强大的语义表示能力和广泛的适用性成为自然语言处理领域的热门选择，但在使用过程中需要注意一些调参和数据质量方面的考虑。

1. **相似度算法间的比较**
   * + 1. 欧几里得距离和曼哈顿距离

欧几里得距离：

用于测量多维空间中两点之间距离的方法。基于欧几里得几何中的距离概念。算法计算两个点之间的直线距离，即两点间的最短路径。

曼哈顿距离：

用于测量多维空间中两点之间距离的方法。曼哈顿距离是通过将两点在每个维度上的坐标差的绝对值相加来计算的，而不是直线距离。

优点：

* 数学简单，适用于处理数值型数据。

缺点：

* 对异常值敏感，因为它以绝对距离为基础，极端值可能对结果产生较大的影响。
* 在高维空间中的计算可能不够准确。
* 对于非数值型数据，如文本或分类数据，这些算法通常不适用。
  + - 1. JACCARD相似系数

用于衡量两个集合之间相似性的统计度量。它通过计算两个集合的交集元素数量除以它们的并集元素数量来得出相似性指数。Jaccard相似系数范围从0到1，其中0表示完全不相似，而1表示完全相似。

优点：

适用于处理文本或集合型数据，如文档的词汇集或用户的兴趣集。

对于处理稀疏数据非常有效。

它也不受数据尺度的影响，对于不同数据的缩放非常稳健。

缺点：

忽略了元素的数量信息，只关注元素的存在或缺失，这可能导致不同集合大小之间的偏见。

通常适用于处理离散数据，而对于连续性数据可能需要进行额外的处理或转化。

* + - 1. 余弦相似度

用于度量两个向量之间的夹角和方向的相似性的方法。余弦相似度计算两个向量之间的夹角余弦值，其中1表示完全相似，0表示没有相似性，-1表示完全相反。这一度量方式特别适用于处理高维数据，因为它不受维度的影响，更多地关注向量之间的方向。

优点：

* 适用于处理文本文档或向量表示的词汇
* 不受数据的尺度影响，对于不同数据的缩放非常稳健。
* 可以处理高维数据

缺点：

* 余弦相似度假定向量是线性的，如果向量之间存在非线性关系，则余弦相似度可能不适用。
* 在处理稀疏数据时可能存在问题，因为余弦相似度只考虑用户评分而忽略了其他信息量的弊端。

1. **其他文本特征提取方法：**

TF-IDF（Term Frequency- Inverse Document Frequency）

TF-IDF，中文叫词频-逆文档频率，用于衡量一个词语在文档集合中的重要性。词频指的是一个词语在文档中出现的频率。逆文档频率的计算方式是通过文档集合中包含某个词语的文档数目的倒数来表示。TF-IDF对于一个文档中的词语，如果它在当前文档中频繁出现（高词频），同时在整个文档集合中出现的文档数较少（高逆文档频率），那么它的TF-IDF权重将较高。这种权重计算方法有助于找到文档中的关键词语。

TF-IDF可以通过sklearn库来实现，即通过调用sklearn.feature\_extraction.text.TdidfVectorizer。调用TfidfVectorizer()会返回一个对象，一旦该对象被拟合到训练集的文本数据，它将返回一个TF-IDF权重矩阵，其中每行表示一个文档（或文本样本），每列表示一个单词（或特征）。矩阵中的每个元素表示对应文档中对应单词的TF-IDF权重。以下是TF-IDF的优点和缺点：

优点：

* 解释性强： TF-IDF得分直观，容易理解，因为它基于词频和逆文档频率的简单数学计算。
* 重要性衡量： TF-IDF能够捕捉单词在文档中的重要性，通过词频和逆文档频率的组合，更好地反映了单词在文档集合中的信息量。
* 去除常见词： TF-IDF可以有效去除常见的停用词，因为这些词通常在各种文档中频繁出现，其TF-IDF得分较低，从而不会干扰分析和模型的性能。

缺点：

* 忽略词序信息： TF-IDF不考虑单词的顺序和语法结构，因此忽略了词语之间的语义关系。
* 对长文本不敏感： 在处理长文本时，TF-IDF可能会受到文档长度的影响，较长的文档通常会有更高的词频。这可能导致较长文本中的重要信息不被充分捕捉。
* 依赖文档集合： TF-IDF的性能取决于文档集合，如果文档集合不充分或者过于庞大，逆文档频率的估计可能不准确。

主要内容：

1. 罗列各个算法优缺点，实现难度。

2. 根据本项目现有要求和条件对各个算法进行可行性分析。

## 拟使用算法

1. Word2vec + 余弦相似度和tf-idf

首先对训练集进行数据预处理，然后调用Word2vec和Tfidfvectorizer来计算训练集的句子向量。然后通过余弦相似度计算验证集里的数据和标准数据之间的相似度。将相似度最高的句子放入结果列表里。如果次高相似度比最高相似度的百分差异小10%且大于某一个指定的距离，则也将该句子放入结果列表里。但结果列表里句子数量不能超过某一个值。如果句子中的单词都不在 tf-idf 和词嵌入模型中的词汇表里，则输出”no\_similar\_sentence”。最后计算预测答案的精确率，召回率和f1值。

因中文电子病历的描述涉及到很多特定术语，使用已训练好的网上大规模Word2Vec模型效果不太好。需要自行训练领域特定的Word2Vec模型。经反复调整Word2vec模型的形参（可在”model performance.txt”查看不同形参生成的模型的效果），可得效果最好的Word2Vec模型是第10版模型，形参为vector\_size=300, window=8, min\_count=1, epochs=2。

Word2Vec通常基于上下文窗口的共现关系来学习单词的嵌入向量。在医学领域，某些术语的语义可能高度依赖于特定的医学上下文，这可能导致模型在一般性的语义相似度计算上失效。除此之外，Word2Vec可能难以处理词义歧义，即一个词汇具有多个不同的含义。在医学领域，这种问题可能更加显著，因为很多医学术语具有多种不同的用途。而且，Word2Vec和余弦相似度是黑盒模型，通常难以提供详细的解释或理解为什么两个词汇被认为相似。在医学领域，可解释性通常很重要。实验表明使用此模型完成本任务，只能达到0.25%的f1值。

1. 无微调BERT

本次作业采用的BERT模型是经预训练的Bert\_base\_chinese模型，相对于BERT的其他变种来说规模较小，参数数量相对较少，适用于一些资源受限的环境。尽管规模较小，BERT\_base\_chinese仍然能够提供强大的语义表示能力，使其在中文自然语言处理任务中表现出色。

首先对训练集作预处理，包括去重、去除非中文字符、分词等，通过建立Dataset类来对文本进行tokenizer处理并规范化输出，再调用bert\_base\_chinese模型来计算匹配目标文本的词嵌入后保存，最后将测试集文本经相同处理后得到的词嵌入二维化处理，与保存好的目标文本的特征值做一对多相似度匹配，相似度计算方法采用余弦相似度。

由于BERT模型对硬件要求较高，采用微调处理并对xls文件中的标准词作一对多或多对多匹配会花费巨量时间，因此为了提高作业进度推进效率，报告中仅介绍无微调的结果。

## 主要步骤介绍

步骤介绍中省略掉导入库的过程，库的介绍放在readme文件中。本大作业中LEONARDO同学负责Word2vec+余弦相似度和tf-idf的工作，郑安玮负责无微调BERT的所有工作。

* + - 1. **Word2vec + 余弦相似度和tf-idf:**
      2. 将train.json中的text和“normalized\_result”以及dev.json中的“text”数据预处理。（对“text”数据进行分词，去重和去除停用词，对”normalized\_result”进行分词，去重和去除标点符号）
      3. 调用Gensim库里的Word2vec，使用第一步生成的数据为实参。返回为基于该数据的词嵌入模型。
      4. 调用sklearn库里的TfidfVectorizer生成一个对象。将该对象拟合第一步生成的数据。返回为基于该数据的TF-IDF权重矩阵。
      5. 通过该词嵌入模型算出train.json中的“normalized\_result”数据和dev.json中的”test”数据的每个句子的句子向量。具体步骤如下：

1. 通过词嵌入模型计算句子中的每个词的词向量
2. 通过Tfidf权重矩阵计算句子中的每个词的权重
3. 将句子中的每个词的词向量跟其对应的权重相乘
4. 将句子中加权后的所有词向量相加，并除以句子的个数（取平均）
5. 该向量即为句子的句子向量。
   * + 1. 枚举dev.json中的“test”数据的句子向量，使用余项相似度计算该句子向量与train.json中的”normalized result”的所有句子向量的相似度。
       2. 将一个相似度最高的句子放入结果列表里。如果次高相似度比最高相似度的百分差异小10%且大于某一个指定的距离，则也将该句子放入结果列表里。但结果列表里句子数量不能超过某一个值。实验表明当距离为0，且最大数量为2时，得到的效果最好。如果句子中的单词都不在 tf-idf 和词嵌入模型中的词汇表里，则输出”no\_similar\_sentence”。
       3. 按第六步骤列举完dev.json中的“test”数据的句子向量，得到该模型的预测答案列表。
       4. 对比该预测答案与dev.json中的”normalized\_result”数据，计算预测答案的真阳性数量，预测总数，和答案总数。
       5. 计算预测答案的精确率，召回率和f1值。公式如下：

精确率=真阳性数量/预测总数

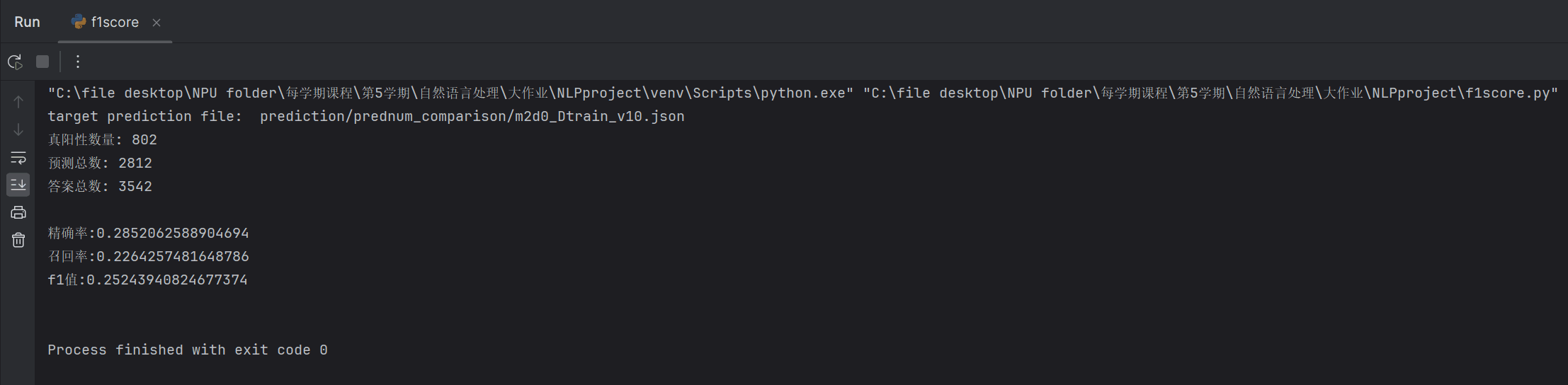
召回率=真阳性数量/答案总数

f1值=2\*精确率\*召回率/(精确率+召回率)

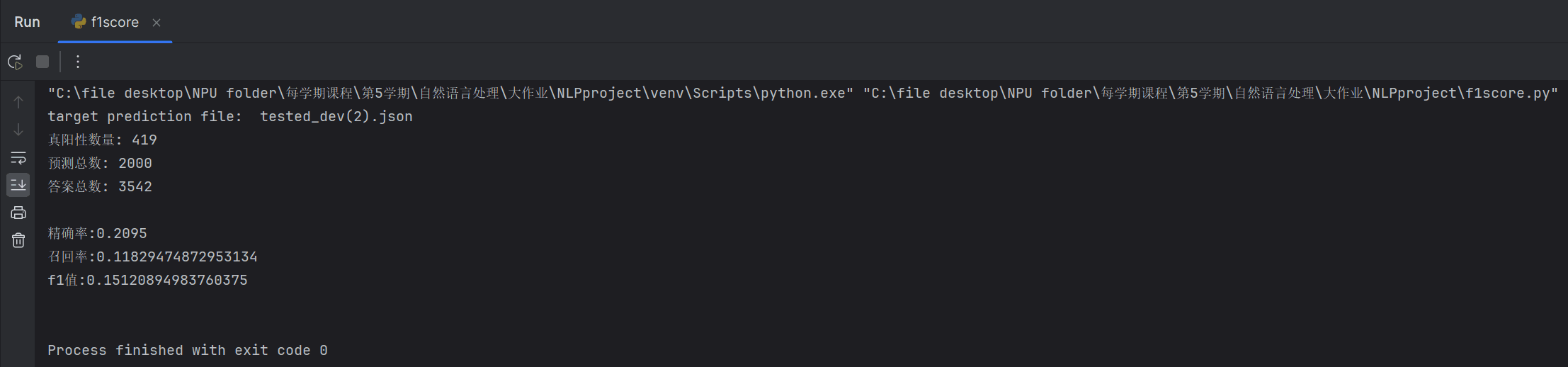
* + - 1. **无微调BERT**
      2. 将train.json中的text和“normalized\_result”以及dev.json中的“text”数据预处理。（对“text”数据进行分词，去重和去除停用词，对”normalized\_result”进行分词，去重和去除标点符号）
      3. 实例化本地bert\_base\_chinese模型中的tokenizer与model以便后续使用。
      4. 计算normalized\_result的词嵌入，并将最后结果存入csv以便后续使用。
      5. 选择dev.json作为测试集，对文件中的text做相同处理后，与预存的csv文件中的normalied\_result计算预先相似度并排序，将相似度最高的结果存入列表。
      6. 将text与匹配所得的目标文本列表拼接为字典，生成tested.json文件，用于计算F1。
      7. F1的计算过程与Word2Vec+预先相似度与TF-IDF计算方法相同。

## 计算F1结果截图

**Word2Vec/TF-IDF**



**BERT**



## 项目总结评估及未来发展展望

从两组模型与算法计算得出的F1指数可以发现，两种方法的效果都并不理想。

Word2vec + 余弦相似度和TF-IDF的F1结果仅有0.25左右。在医学领域，某些术语的语义可能高度依赖于特定的医学上下文，这可能导致模型在一般性的语义相似度计算上失效。

而且很多医学术语具有多种不同的用途，Word2Vec可能难以处理这些词义歧义。

无微调BERT的F1结果仅有0.15左右，这一方面与预训练模型固有的领域差异、模型与数据结构不同有一定关系，因此若能够实现对模型的微调，利用train.json对模型进行强化训练，匹配效果应该会有所上升。

这次课程项目开发促使我们学习了大量的NLP领域的预处理方法、模型应用与算法，帮助我们认识到了有关模型架构与数据结构的新知识，也初次了解到不同的模型尤其是复杂模型对计算机硬件的要求。

NLP从上世纪50年代开始发展以来，各种技术的出现都推动了这一领域的进步，包括深度学习、词向量与词嵌入、预训练模型、迁移学习等，尤其在20世纪，NLP依托硬件的进步得到了更大的算力支持。未来NLP发展趋势包括更大规模的模型、多模态融合、自监督学习、领域自适应、实时和在线学习、更好的模型解释性、个性化和情感分析、以及伦理和法律方面的考虑。这些方向将推动NLP在语言理解、交互和应用方面取得更多创新。