LLM—BLENDER:利用成对排序和生成融合组建大型语言模型

太长不看版：

本文介绍了一个名为LLM-BLENDER的集成框架，旨在通过结合多个开源大型语言模型（LLM）来提升性能。LLM-BLENDER由两个核心模块组成：PAIRRANKER和GENFUSER。PAIRRANKER通过成对比较方法来区分不同LLM生成的候选输出之间的细微差异，而GENFUSER则负责合并排名靠前的候选项，以产生更优的输出。

研究者们创建了一个名为mixdirective的基准数据集，该数据集混合了多个指令数据集，并采用了oracle两两比较的方法。实验结果表明，LLM-BLENDER在多个指标上都明显优于单个LLM和基准方法，显示出其在性能上的实质性提升。

LLM-BLENDER的提出基于开源LLM的多样性和互补性，旨在通过集成学习的方法，提高输出的鲁棒性、泛化能力和准确性。研究结果对于开发者和研究人员在部署和研究集成学习的LLM方面具有重要意义。此外，本文还探讨了未来可能的发展方向，包括扩展框架以支持更多类型的模型、开发更复杂的排名和融合技术，以及探索减少计算开销和快速适应新领域的策略。总体而言，LLM-BLENDER强调了结合多个模型的独特贡献对于推动人工智能发展的价值。

摘要：

本文提出了LLM-BLENDER集成框架，通过利多个开源大型语言模型(LLM)来获得卓越性能。框架由两个模块组成:PAIRRANKER和GENFUSER，解决了不同示例的最佳LLM可能存在的问题。

PAIRRANKER采用专门的两两比较方法来区分候选输出之间的细微差异。它对输入文本和一对候选文本进行联合编码，使用交叉注意编码器来确定较优的文本。研究结果表明，PAIRRANKER与基于ChatGPT的排名具有最高的相关性。

GENFUSER的目标是合并排名靠前的候选对象，通过利用它们的优点和减轻它们的缺点来产生改进的输出。

为了便于评估，引入了一个基准数据集mixdirective，这是多个指令数据集的混合，具有oracle两两比较的特点。LLM-BLENDER在各种指标上明显优于单个LLM和基准方法，有着实质性的性能差距。

引言：

大型语言模型(在各种任务中表现出令人印象深刻的性能，主要是由于它们能够遵循指令并访问大量高质量数据，这展现了人工通用智能的未来。然而，著名的大语言模型，如GPT-4和PaLM 是不开源的，限制了我们对其架构和训练数据的了解。开源的模型如Pythia、LLaMA和Flan-T5提供了在定制指令数据集上微调这些模型的机会，从而能够开发更小但更高效的LLM，如Alpaca、Vicuna、OpenAssistan和MPT。由于数据、体系结构和超参数的变化，开源LLM表现出不同的优点和缺点，使它们相互补充。

图1展示了我们收集的5000条指令的最佳LLM分布。虽然Alpaca的比例最高，但仅占21.22%。此外，饼状图表明，不同示例的最佳LLM可能会有很大差异，而且没有开源的LLM在竞争中占主导地位。因此，重要的是动态集成这些LLM，以便为每个输入生成一致的更好的响应。考虑到LLM的不同优势和劣势，开发一种集成方法来利用它们的互补潜力，从而提高鲁棒性、泛化和准确性。

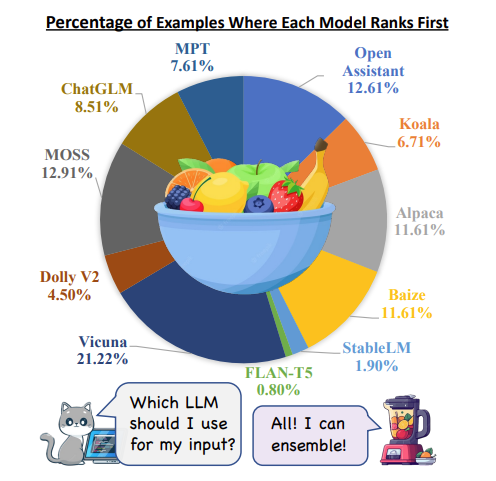


图1:大语言模型互组的占比，根据这个关于每个LLM排名第一的示例百分比的饼状图，我们可以看到不同示例的最优LLM可能有很大差异

LM-BLENDER包含两个模块:PAIRRANKER和GENFUSER。最初，PAIRRANKER比较N个LLM的输出，然后GENFUSER融合这些输出，从排名前K的输出中生成最终输出。现有方法，包括InstructGPT中的奖励模型用于对输出来自给定输入x的语言模型，主要侧重于基于x单独对每个进行评分，采用的编码模块。虽然在候选人差异明显的情况下，这种排序目标可能是强大而有效的，但在整合大语言模型时，它可能不那么有效。在大语言模型的候选输出中，候选输出的差异非常细微，因为它们都是由非常复杂的模型产生的，一个可能只比另一个好一点点。因此，我们提出了一种专门的两两比较方法，PAIRRANKER，以有效地识别候选输出之间的细微差异并提高排名性能。特别是，我们首先为每个输入收集N个模型的输出(例如，图1中的N = 11个模型)，然后创建N(N−1)/2对输出。我们将输入x和两个候选输出和作为交叉注意编码器(例如RoBERTa (Liu et al, 2019))的输入，以的形式共同编码，以学习并确定哪个候选更好。

在推理阶段，我们计算一个包含对数的矩阵，表示两两比较的结果。给定这个矩阵，我们可以推断给定输入x的N个输出的排序。随后，我们可以对每个输入使用PAIRRANKER中排名靠前的候选者作为最终结果。因此，这种方法不依赖于所有示例的单一模型;相反，PAIRRANKER通过综合比较所有候选对来为每个示例选择最佳模型。

尽管如此，这种方法可能会限制产生比现有候选更好产出的潜力。为了研究这种可能性，引入了GENFUSER模块来融合N个排名的候选对象中的前K个，并为最终用户生成改进的输出。我们的目标是利用前K名候选人的优势，同时消除他们的弱点。”

为了评估LLM集成方法的有效性，我们引入了一个名为mixdirective的基准数据集(第2.2节)。在这个数据集中，我们使用N=11个流行的开源LLM，为各种现有的指令遵循任务(格式为self - struct)中的每个输入生成N个候选项。该数据集包括100k个训练样例和5k个验证样例，用于训练候选排名模块(如PAIRRANKER)，以及5k个带有oracle比较的测试样例，用于自动评估。

我们对mixdirective基准的实证结果表明，LLM-BLENDER框架通过集成LLM显著提高了整体性能。PAIRRANKER做出的选择优于任何固定的单个LLM模型，正如在基于参考的指标和GPT-Rank中的优越性能所表明的那样。通过利用PAIRRANKER的顶级选择，GENFUSER通过有效融合到最终输出进一步提高响应质量。LLM-BLENDER在传统指标(即BERTScore, BARTScore, BLUERT)和基于chatgpt的排名方面都取得了最高分。LLM- bl E N D E R在12种方法中的平均排名为3.2，明显好于最佳LLM的3.90。LLM-BL E N D E R的产出排在前3名，占68.59%，而Viccuna仅达到52.88%。我们相信LLM-BL E N D E R和我们的研究结果将有利于从业者和研究人员部署和研究集成学习的LLM。

Preliminaries（正文前书页）：

我们首先提供了问题的表述和两种常见的集成方法。接下来,我们展示为训练和评估目的而创建的数据集MixInstruct。最后，我们概述了我们的框架。

# 2.1问题的建立：

给定输入x和N个模型，，我们可以通过对每个模型处理x来生成N个候选输出。我们用Y = 。在训练数据中，我们假设有一个ground truth输出y，而它在测试时评估时仍然是隐藏的。

在实践中，人们可能会选择一个固定的模型，比如，来推断所有未见过的例子(即，总是使用作为x的最终输出)。如果在某些观察到的例子上表现出明显更好的整体性能，这可能是合理的。然而，依赖于预先选择的模型可能会导致次优性能，因为N个模型在不同情况下可能具有不同的优点和缺点，这意味着不同x值的最优选择可能并不总是来自同一模型。我们的目标是开发一种集成学习方法，为输入x产生输出，最大化相似度Q(, y; x)

具体来说，给定一个测试集，我们的目标是最大化。

集成大语言模型有两种主要方法:基于选择的方法和基于生成的方法。基于选择的方法比较集合Y中的候选对象，选择排名最高的候选对象。这意味着，∈Y。由于选择的固有性质和有限的解空间，基于选择的方法的性能受到所考虑的N个候选者的限制。相反，基于生成的方法侧重于从Y中融合K个候选(1 < K≤N)，以产生一个看不见的响应作为最终输出。

# 2.2 mixdirective:一个新的基准：

引入了一个新数据集mixdirective，用于测试LLM在指令跟随任务中的集成模型。主要从四个来源收集了一组大规模的指令示例，如表1所示。

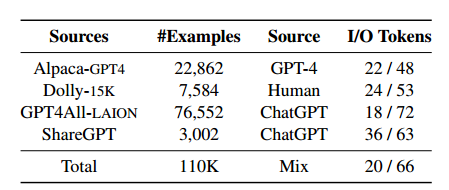


表1:mixdirective的统计数据。它包含110K个示例，我们将数据集随机分为100K/5K/5K大小的train/dev/test。

在整理和处理这些开源数据之后，我们采样了100k个示例用于训练，5k个用于验证，5k个用于测试。然后，我们在这11万个示例上运行N = 11个流行的开源LLM，包括Vicuna、OpenAssistant、Alpaca、MPT等(参见表2和图1)。

为了获得候选的oracle排序，我们设计了ChatGPT的比较提示来评估所有候选对。具体来说，每次考试-我们准备了55对候选体(11 × 10/2)。对于每一对，我们要求ChatGPT判断更好的候选人(或宣布平局)。提示模板可以在附录中找到。对于训练集和验证集，我们提供了基于传统指标(如BERTScore、BLEURT和BARTScore)的结果。在这种情况下，我们使用函数Q(, y)根据其与基本真理y的相似性来估计候选的质量

# 2.3 LLM-BLENDER：一个新颖的框架

LLM-BLENDER，用于集成大语言模型，如图2所示。该框架由两个主要组件组成:配对排序模块PAIRRANKER(第3节)和融合模块GENFUSER(第4节)。PAIRRANKER模块学习比较每个输入的所有候选对，并随后对候选列表进行排序。然后，我们选择排名前K = 3的候选项，将它们与输入x连接起来，并构建GENFUSER模块的输入序列。GENFUSER模块是一个seq2seq LM，它最终生成为用户服务的最终输出。

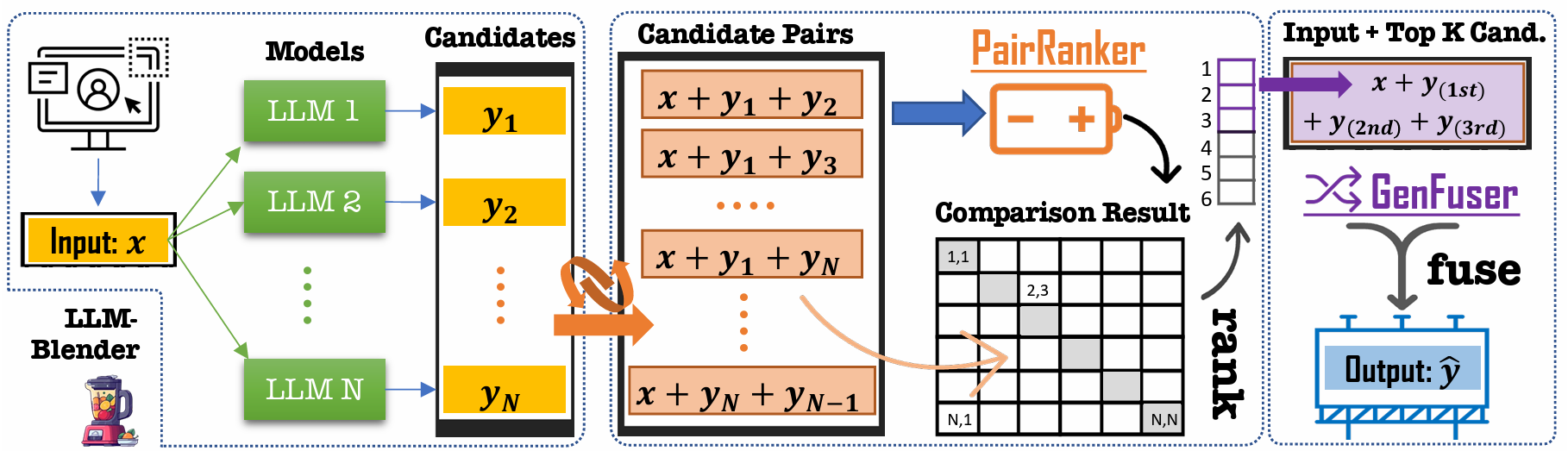


图2:LLM-BLEND E R框架。对于来自用户的每个输入x，我们使用N个不同的LLM来获得候选输出。然后，我们将所有候选者配对，并将它们与输入连接起来，然后将它们提供给PAIRRANKER，生成一个矩阵作为比较结果。通过将结果聚合到矩阵中，我们可以对所有候选对象进行排序，并取其中的前K进行生成融合。GENFUSER模块将输入x与K个排名靠前的候选人连接起来作为输入，并生成最终输出y´。

PAIRRANKER: 成对的排名

在本节中，我们介绍了3.1节中对Y中的候选对象进行排序的三种基准方法，并提出了提出的PAIRRANKER方法。

# 基线方法

以往的重排序方法主要是独立计算每个候选∈Y的分数 = ，其中仅由决定。值得注意的是，GPT-3.5指令调优中的奖励模型也属于这一类。图3说明了这些基线方法，下面的段落将进一步详细说明。

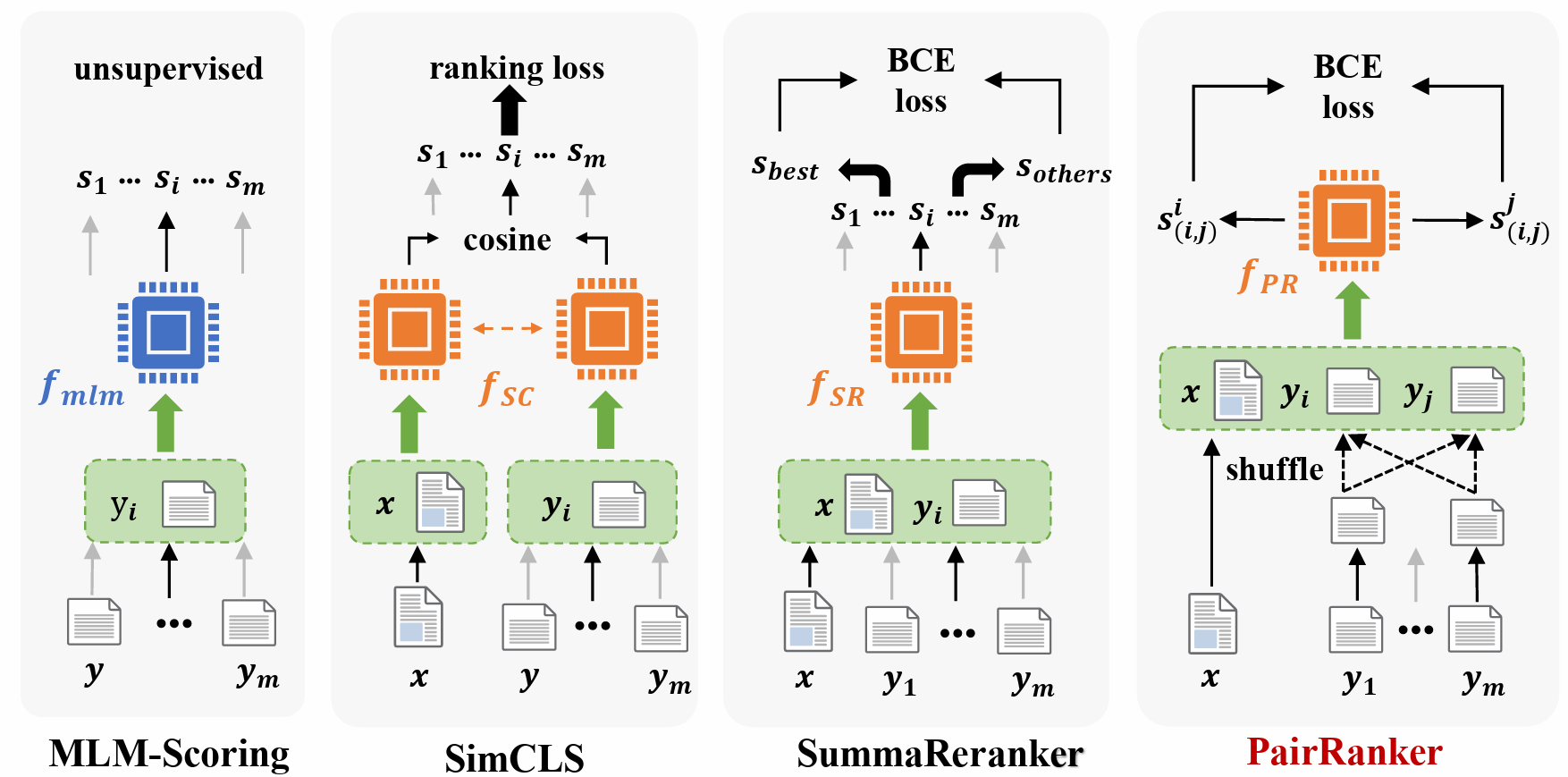


图3:典型的重新排序方法的体系结构。X是一个输入，是一个确定的候选项，它的分数是。mlm评分是一种无监督的方法，使用外部屏蔽LM对候选人进行评分;SimCLS使用相同的编码器对x和每个候选进行编码;取而代之的是，SummaReranker使用交叉编码器同时对x和进行编码;PAIRRANKER同时对一对候选人(, )进行编码，对其进行配对评分，生成每位候选人的最终分数，如图4所示。

## MLM-Scoring

通过计算其伪对数似然来评估候选对象的质量，伪对数似然是通过一个接一个地屏蔽令牌并使用屏蔽LMs(例如BERT)计算被屏蔽令牌的对数似然来获得的。给定一个候选作为单词序列，则伪对数似然为:。这种无监督的方法对于机器翻译和语音识别等NLG任务的输出重新排序是有效的。

## SimCLS

对输入x和每个生成的候选∈Y 进行编码. 使用相同的编码器H，得到H(x)和H()。它们之间的余弦相似度 = cos (H(x)， H())作为预测分数，因为H(x)和H()共享由语言编码器诱导的相同嵌入空间。在训练中，利用边际排序损失来优化H。

## SummaReranker

将输入x和每个候选连接起来，使用交叉注意编码器来学习排名。具体来说，它们使用H([x;])来预测分数，其中H是一个Transformer模型。在训练阶段，利用二值交叉熵(BCE)损失来区分最佳候选。

## 局限性：

尽管使用对比损失，但这些方法依赖于个体得分进行推理。编码器没有暴露在成对的候选对象中进行直接比较学习。这种点式排名方法可能不足以在大语言模型和指令遵循任务的背景下选择最佳候选人。一个原因是，当选择的大语言模型受欢迎且具有竞争力时，产出的质量通常很高。此外，与总结任务不同，指令任务的回答可能是开放式的。因此，仅仅考察个别候选人无法得出可靠的分数。对于较短的回答，这个问题变得更加突出，其中序列可能只有几个单词不同，但在有益，有害和公平方面差异很大。鉴于这些限制，我们认为个人评分方法可能无法捕捉到关键的细微差别。

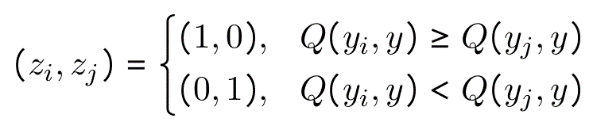
# 3.2 两两比较

为了解决点向排序的局限性，我们的目标是训练一个带有参数φ的排序器f，它可以通过将一对输出候选对象与输入文本一起编码来比较它们。我们的排名模块应该专注于学习，以捕捉两个候选人之间的差异，并选择质量更高的候选人。给定一对候选对象, ，我们得到它们的配对特定分数:和。我们将模型认为优于的置信度记为=:−。我们可以使用这些分数来推断从Y导出的所有配对的最终排名。

为了学习这种能力，我们将输入x和两个候选项连接起来形成一个序列[x;;]并将其送入交叉注意变压器，得到特征: ([x;;])进行建模。我们假设有多个Q函数来优化，如BERTScore, BARTScore等，并将学习问题视为一个多任务分类问题:

，

其中σ表示sigmoid函数，

.

为了对多个Q优化，我们取最终多目标损失的平均值:L =∑

# 3.3 PAIRRANKER架构：

本小节讨论了PAIRRANKER模块的具体设计。

## 编码：

我们使用Transformer层对一个输入和一对候选对象进行编码，使注意力能够在输入的上下文中捕获候选对象之间的差异。我们按顺序连接这三个片段，并使用特殊标记作为分隔符形成单个输入序列:<source>、<candidate1>和<candidate2>。生成的transformer输入序列的形式为“<s><source> x </s> <candidate1> </s> <candidate2> </s>”，其中x是源输入的文本，和是两个输出候选者的文本。特殊标记<source>、<candidate1>和<candidate2>的嵌入分别用作x、和的表示，表示为x、、。

## 训练：

为了确定两个候选的分数，我们分别将x与和的嵌入连接起来，并将它们传递给单头层，这是一个多层感知器，最后一层的维度等于要优化的Q函数的数量。此维度中的每个值表示特定Q函数的计算Q分数。我们通过平均这Q个分数得出候选人的最终分数或。由于存在O()个唯一对组合，我们在训练阶段采用了有效的子采样策略来保证学习效率。

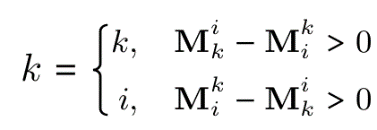
在训练过程中，我们从候选池中随机选择一些组合，而不是所有的N(N−1)/2对。我们还将目标文本与其他候选文本进行比较，方法是通过将基本事实y混合到y中来扩展候选文本池。在实践中，我们发现每个输入使用5对就足以获得不错的结果。由于语言模型的位置嵌入，一对(x, , )中的候选项的顺序很重要，作为的比较结果，(x, , )和(x, , )可能不一致。因此，我们对每个训练对中的候选序列进行洗牌，以便模型学习与自身保持一致。

## 推论：

在推理阶段，我们对每一对候选(, )∈y2获得分数。经过N(N−1)次迭代，我们得到一个矩阵M，其中 = 表示优于 的置信度。为了识别基于M的最佳候选者，我们引入了三个聚合函数来确定Y的最终排名。

我们提出了两种评分方法，MaxLogits和MaxWins，它们利用了矩阵中的所有元素。设和表示矩阵的第i行和第j列。对于每个候选，其MaxLogits分数定义为 =∑(−)，而其MaxWins分数定义为 =∣{∈∣ > 0}∣+∣{∈∣ < 0}∣，其中∣∣表示集合大小。

本质上，MaxLogits计算优于所有其他候选人的置信度，而MaxWins计算与其他候选人比较的胜利次数。然而，这两种方法需要对N个候选对象进行O()次迭代，这可能会造成计算负担。因此，我们提出了一种更有效的聚合方法，执行单个气泡排序运行与两两比较，以选择最佳候选人。我们首先对Y中的候选索引顺序进行洗牌以获得默认顺序，并将最佳候选索引k初始化为1。我们迭代更新最佳候选索引如下:



经过N−1次比较，我们选择yk作为最佳候选者。该方法将推理时间复杂度从O()降低到O(N)，与以往的点向方法保持一致。

不管聚合方法是什么，我们都可以对所有候选项进行y排序。我们的实验(见附录)表明MaxLogits产生了最好的性能，因此我们使用MaxLogits作为PAIRRANKER的默认聚合器。

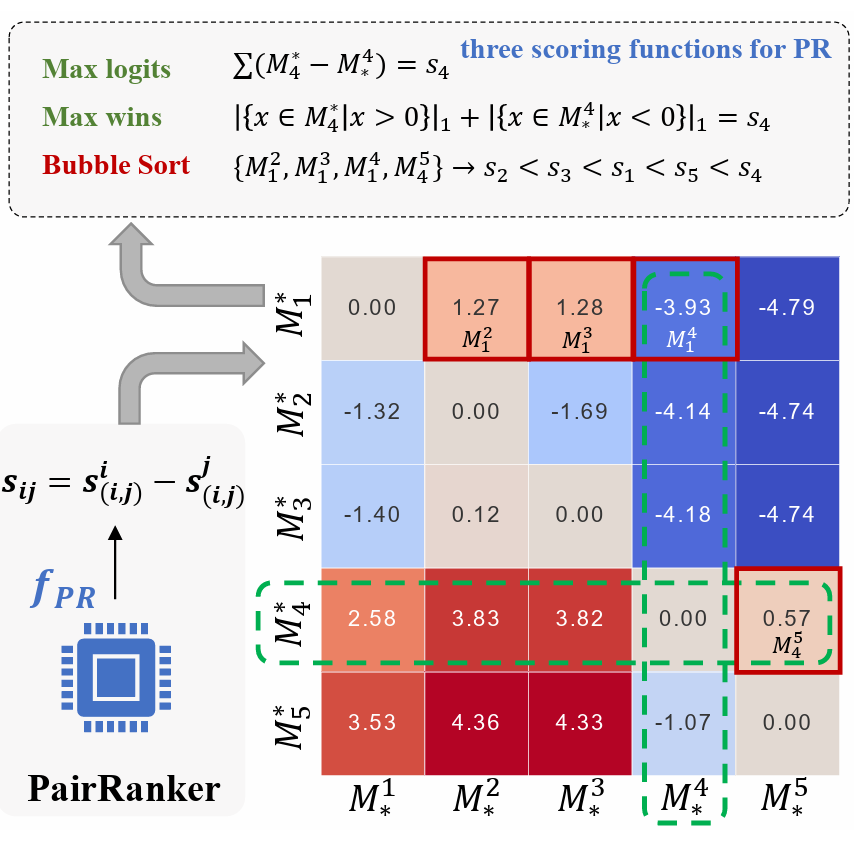


图4：Pairranker的集合方法。

生成融合：

PAIRRANKER的有效性受到候选库y中选择的质量的限制。我们假设通过合并多个排名靠前的候选，我们可以克服这一点约束。由于这些优秀的候选人往往展示出互补的优势和劣势，因此结合他们的优势，同时减轻他们的缺点，从而产生更好的回应是合理的。我们的目标是设计一个生成模型，该模型接受输入x和K个排名靠前的候选项(例如，K = 3)，并产生改进的输出Y´作为最终响应。

为了实现这一点，我们提出了GENFUSER，这是一种seq2seq方法，用于融合一组以输入指令为条件的候选指令，以生成增强的输出。具体来说，我们使用分隔符(如<extra\_id\_i>)顺序连接输入和K个候选项，并微调类似t5的模型以学习生成y。在实践中，我们使用Flan-T5-XL (Chung et al, 2022)，由于其优越的性能和相对较小的尺寸，它具有3b个参数。

评价：

# 5.1 设立：

使用mixdirective(第2.2节)进行评估。

# NLG指标：

我们使用两种类型的评估度量(例如，Q)。第一组是NLG任务的传统自动指标:BERTScore)、BLEURT和BARTScore

# GPT-Rank：

第二种是基于提示ChatGPT对所有候选人进行两两比较，并根据获胜次数决定他们的排名(即MaxWins聚合)。我们将这个基于gpt的排名指标命名为GPT-Rank。

# Model training:

我们使用DeBERTa (He et al, 2021) (400m)作为PAIRRANKER的主干，GENFUSER基于Flan-T5-XL (3b)。根据我们的消融研究，我们选择使用BARTScore，因为它与GPT-Rank的相关性较好，见5.2。

# 5.2主要结果：

在表2中，我们给出了N=11 LLM以及其他方法在mixdirective上的总体性能。除了三个自动指标和GPT-Rank之外，我们还显示了每种方法在GPT-Rank方面产生的输出优于或与两个顶级LLM(即OpenAssistant(≥OA)和Vicuna(≥Vic))相同的示例百分比。

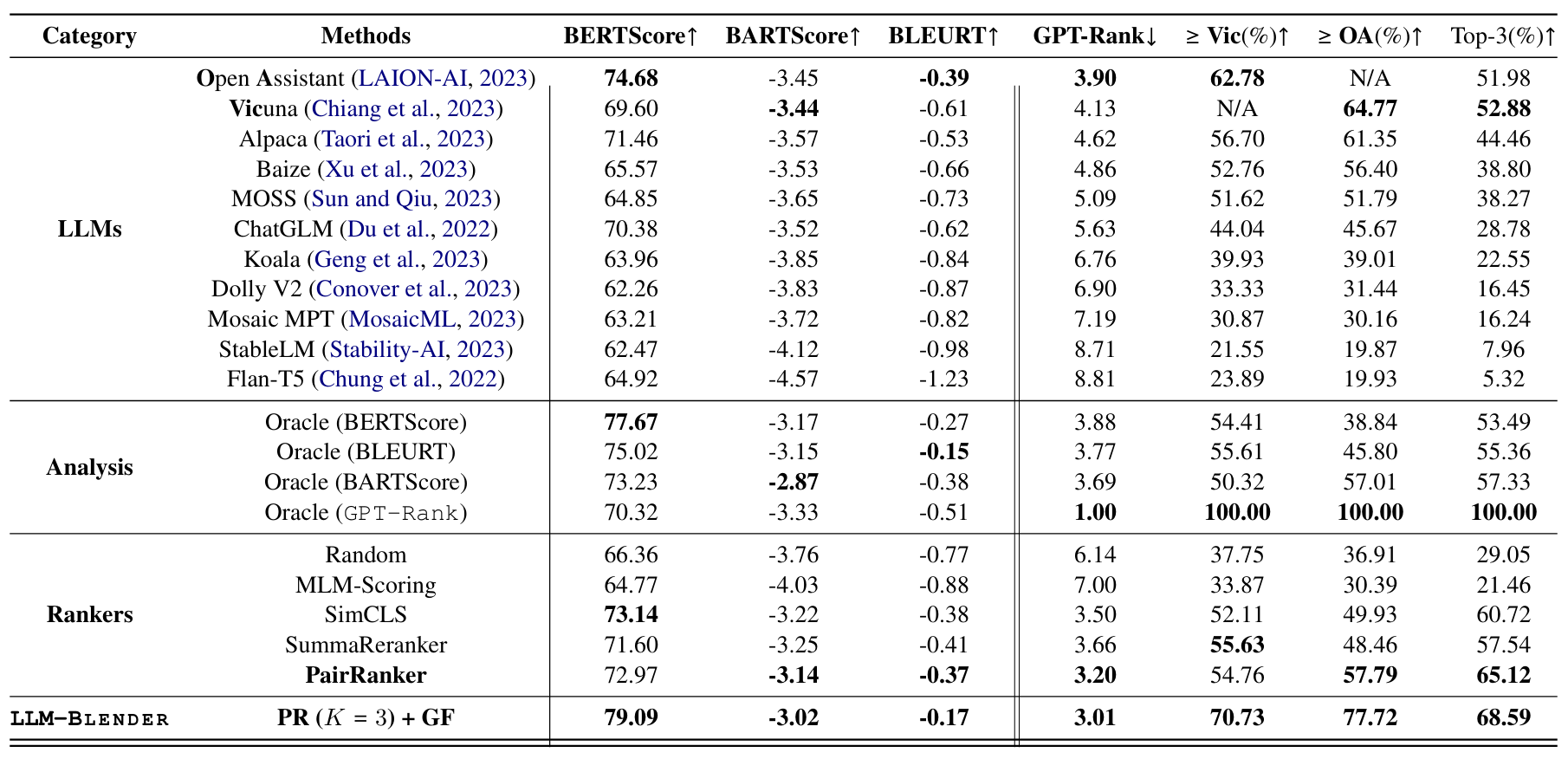


表2:mixdirective的实证结果。GPT-Rank是最重要的指标。

# 大语言模型之间有不同的优势和劣势。

该表根据ChatGPT (GPT-Rank)确定的平均排名按排序顺序显示LLMs。其中，Open Assistant、Vicuna和Alpaca是排名前三的模型。继他们之后，三个著名的LLM，即Baize, Moss和ChatGLM，已经使用中英文教学数据进行了微调，也在MixInstruct上表现出色。相反，Mosaic MPT、StableLM和Flan-T5在评价中排名倒数3位。然而，最高/最低模型的平均gpt排名与第一/最后一名(1或11)保持着明显的距离，这凸显了大语言模型组合的重要性。

# 顶尖的大语言模型并不总是优秀的。

很明显，尽管OA和Vic表现得非常好，但仍有相当大比例的例子表明，其他大语言模型的表现被认为优于它们。例如，尽管Koala的平均GPT-Rank为6.76，但大约40%的例子表明，Koala产生的反应比OA和Vic更好或同样好。这进一步强调了采用我们的LLM-BLENDER框架进行排名和融合的重要性。

# NLG指标：

此外，我们根据每个指标本身对oracle (top-1)选择的性能进行了全面的分析。研究结果表明，这些选择在其他指标上也表现出良好的性能。例如，从GPT-Rank中得到的oracle选择实现了−3.33的BARTScore，超过了OA(−3.45)。相反，BARTScore的oracle选择在GPT-Rank上的收益率为3.69，也显著优于OA(3.90)。这一观察结果证实了使用BARTScore对PAIRRANKER进行监督的合理性，表3也说明了这一点。

# PAIRRANKER优于其他排名：

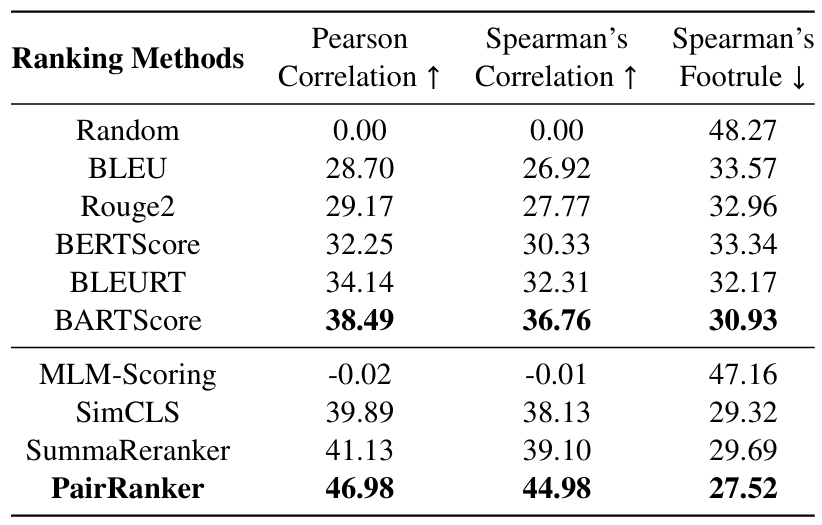
MLM评分甚至不能胜过随机选择，突出其无监督范式的局限性。相反，SimCLS、SummaReranker和PAIRRANKER表在BARTScore和GPT-Rank中表现优异。值得注意的是，PAIRRANKER选择的应答的平均gpt排名(3.20)显著优于最佳模型0.70(相对性能提高18%)，也优于所有其他排名。此外，它在BARTScore(- 3.14)等指标上取得了令人印象深刻的结果，具有实质性的优势。PAIRRANKER的选择分别在54.76%和57.79%的样本上优于或等于Vic/OA，在65.12%的样本上排名前3。

# LLM-BLENDER是最好的：

我们使用从PAIRRANKER中选择的前3名，并将它们作为GENFUSER的候选项。基于这种集成，LLM-BLENDER显示出预期的卓越能力。在GPT-Rank方面，它达到了3.01，大大超过了最佳模型OA(3.90)。BERTScore(79.09)、BARTScore(- 3.02)和BELURT(- 0.17)的得分均比最佳模型分别高出4.41、0.43和0.22，显示出明显的优势。此外，LLM-BLENDER也出色地超越了排名前两位的Vic(70.73)和OA(77.72)，从而弥补了PAIRRANKER的不足。

# 排名的相关性：

除了只关注每个排名前1的选择外，我们还对所有具有GPT-Rank的候选人之间的总体排名相关性进行了全面分析(见表3)。这里使用的相关指标包括Pearson相关系数、Spearman相关系数和Spearman Footrule距离(Diaconis和Graham, 1977)。



表三：每个排名方法和oracle排名（GPT-Rank）之间的相关性

事实证明，与其他指标相比，BARTScore与GPT-Rank的相关性最高，这表明我们可以使用BARTScore为训练提供监督。对于排名者，MLM评分但仍然不如随机排列。另一方面，SummaReranker在Pearson correlation(41.13)和Spearman 's correlation(39.10)方面表现出更好的相关性，而SimCLS在Spearman 's Footrule distance(29.32)方面表现出更好的相关性。值得注意的是，PAIRRANKER在所有相关类型中与GPT-Rank的相关性最高，甚至比BARTScore好得多。

结论和展望未来：

在本文中，我们阐述了利用开源大型语言模型(LLM)的各种优势和弱点的动机，旨在创建一个集成框架，利用它们的互补能力，在各种指令遵循任务上产生一致的卓越结果。通过动态集成LLM，我们的目标是减少单个模型中的偏差、错误和不确定性，从而产生与人类反馈更一致的输出。

我们的主要贡献如下:

•一个新的框架:LLM-BL E N D E R是一种用于对多个LLM的输出进行排名和融合的即时集成学习方法。它由两个模块组成:PAIRRANKER和GENFUSER，这两个模块都是简单而有效的。

•新数据集:mixdirective是一个基准数据集，用于训练和评估LLM集成方法在指令跟随任务上的效果。

•有希望的结果:我们表明，我们的方法可以显著改善各种指标的总体结果，我们的发现表明，这一方向对研究界和实践者都是有希望的。

•工具包:通过开源我们的框架，我们的目标是让其他人更容易利用我们的方法，从而开发更先进的人工智能系统，在各种任务中实现鲁棒性、泛化和更高的准确性。

未来的发展方向：潜在的未来方向包括将LLM-BL E N D E R框架扩展到更多类型的模型甚至非文本模式，开发更复杂的排名和融合技术，并研究我们的集成方法在其他领域和任务中的可转移性。此外，探索最小化计算开销的方法和结合主动学习策略以快速适应新的专业领域和数据源是进一步研究的富有成果的领域。总的来说，我们的工作强调了结合多个模型的独特贡献的价值。