使用句子排序方法提高长文档分类的效率

Prathamesh Kokate[1,3], Mitali Sarnaik[1,3], Manavi Khopade[1,3], and Raviraj Joshi[2,3]
[1]Pune Institute of Computer Technology, Pune
[2]Indian Institute of Technology Madras, Chennai
[3]L3Cube Labs, Pune

Abstract

由于基于 Transformer 的模型，尤其是 BERT 的计算限制，固定输入长度和二次注意力复杂性，使得长文档分类面临挑战。此外，使用完整文档进行分类通常是多余的，因为通常只有一部分句子包含必要的信息。为了解决这个问题，我们提出了一种基于 TF-IDF 的句子排序方法，通过选择信息量最大的内容来提高效率。我们的方法探索了固定数量和基于百分比的句子选择，以及结合归一化 TF-IDF 分数和句子长度的增强评分策略。在长马拉地语新闻文章的 MahaNews LDC 数据集上进行评估，该方法始终优于如首句、尾句和随机句子选择这样的基线。使用 MahaBERT-v2，仅在整体上下文基线的准确性下降 0.33 个百分点的情况下，我们实现了几乎相同的分类精度，同时将输入规模减少了超过 50%，推理延迟减少了 43%。这表明可以在不牺牲性能的情况下实现显著的上下文减少，使该方法在实际长文档分类任务中具有实用性。

1 介绍

长文档分类是自然语言处理（NLP）中的一项基本任务，可应用于对研究论文、法律文件、新闻文章和客户评论的分类。基于 Transformer 的模型，如 BERT 及其变体，已在文本分类中展示了最先进的性能。然而，由于输入大小的限制和指数级的注意力开销（Park et al. [2022]），这些模型在处理冗长文档时面临基本的约束（Zaheer et al. [2020]）。标准 BERT 模型截断冗长的输入，导致信息丢失，或者需要额外的机制，如分层处理，增加了计算开销（Wagh et al. [2021]）。因此，如何在保持分类准确性的同时有效地处理长文本仍然是一个未解决的挑战（Devlin et al. [2018]）。传统方法为解决此问题涉及架构修改，如稀疏注意机制或分层模型。然而，这些方法通常会引入额外的复杂性和计算开销。相比之下，我们提出了一种数据驱动的优化技术，可以在保留重要上下文信息的同时减少输入大小，从而使标准 transformer 模型可以高效地分类长文档（Minaee et al. [2021]）。我们不修改模型架构，而是通过使用排序机制选择最相关的句子来优化输入表示。我们的方法利用基于 TF-IDF 的句子排序来提取关键句子，减少冗余信息并最小化输入长度（Qaiser and Ali [2018]）。通过计算每个句子中每个单词的 TF-IDF 分数，将所述文章中每个句子视作独立文档进行排名（Das and Chakraborty [2020], Kim and Gil [2019]）。句子的总 TF-IDF 分数通过汇总其单词的分数获得，得分最高的句子被分配最高排名（Liu et al. [2018a], Das et al. [2023]）。

图 1 阐述了本文提出的关键思想。图中展示了体育和政治领域的例子，在这些例子中，仅突出显示了语义上最相关的句子。这些被选中的句子富含领域特定的术语和背景信息，足以进行准确的文档分类，从而减少对全文处理的需求。这突显了我们的方法的效率，通过选择关键句子进行分类，仅处理文档的一小部分，从而减少计算负担。

我们探讨了多种选择句子的策略，包括以下内容：

为了评估这些策略的有效性，我们在 MahaNews 数据集（Mittal et al. [2023], Aishwarya et al. [2023]）上进行了广泛的实验，这个数据集是一个以主题分类的冗长玛拉塔新闻文章语料库。我们使用 MahaBERT（marathi-bert-v2）（Joshi [2022]）对数据集的压缩上下文版本进行训练和测试，并比较不同选择方法的分类性能。我们的结果表明，基于 TF-IDF 的排序显著优于更简单的选择策略，例如选择首句、尾句或随机抽样的句子。此外，结合长度感知加权进一步提高了准确性，而上下文压缩在不损害性能的情况下显著减少了推理时间。

通过系统地分析上下文压缩技术，我们的工作为基于变压器模型的长文档分类提供了一种实用且高效的替代方案，相比于对架构进行修改。

2 相关工作

由于长文档中包含的大量信息，直接使用传统的分类模型处理它们往往会导致高计算成本和

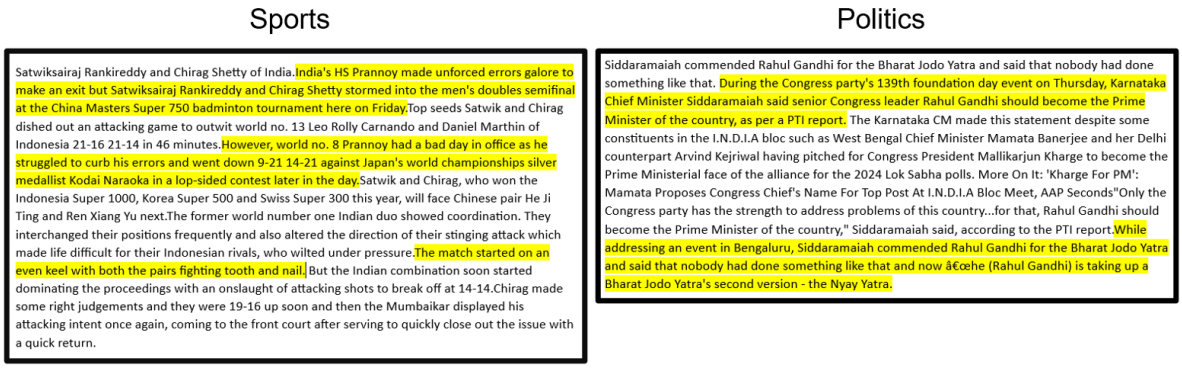

Figure 1: 关键理念的说明——选择性句子处理以实现高效的文档分类。该图展示了两个示例段落：一个与体育相关，另一个与政治相关。在每种情况下，语义上最相关且在上下文中最具信息量的句子被突出显示。这些突出显示的句子包含领域特定的线索（例如，体育活动或政治实体），使得无需处理完整文档即可实现准确分类。这表明选择性句子提取可以在降低计算开销的同时保持分类性能。

增加的推理时间。这就需要提高分类任务的性能。有两种方法可以用来提高长文档分类的效率。它们大致可分为基于数据的方法和基于模型的方法。

2.1 基于模型的方法：

高效处理长文档分类需要在模型复杂性与计算可行性之间取得平衡。为了实现这一点，已经探索了多种技术，包括稀疏注意力机制、量化、递归结构和归一化技术。稀疏注意力机制使得变换器模型能够处理显著更长的输入，同时保留全注意力模型的优势。通过结合用于捕捉整体上下文的全局标记、用于邻近交互的局部标记，以及增强全局覆盖率的随机标记，这些机制有效地将内存和计算成本从二次方降低到线性。这使得它们在长文档分类中特别有用，因为高效处理大量输入序列至关重要。

除了注意力机制之外，降低模型的计算需求也是必不可少的。一个有效的方法是量化，这可以降低模型权重的精度以减少内存使用。例如，Q8 BERT 使用 8 位权重代替标准 32 位，利用了诸如量化感知训练的技术。这显著减少了模型大小，同时保持了高准确性，使其成为在资源受限系统上部署深度学习模型的一个有吸引力的解决方案。

虽然变换器在现代 NLP 任务中占主导地位，但是像长短期记忆（LSTM）网络这样的递归结构也被探索用于捕捉长期依赖关系。LSTM 在保持序列信息方面表现出色，使它们非常适合长文档处理。然而，它们的顺序性质限制了并行化和可扩展性，使基于变换器的模型在更有效地处理大规模文本数据方面有优势。为了进一步增强 Transformer 模型的稳定性和效率，应用了预层归一化。这种技术在注意力机制之前对激活进行归一化，缓解了梯度不稳定性并加速了收敛（Beltagy et al. [2020] ）。通过改善训练动态，预层归一化增强了基于深度 Transformer 架构的稳健性，使其更适合于长文档分类。

通过结合这些技术：稀疏注意力以提高效率，量化以减少计算需求，递归机制以保留序列，以及预层归一化以增强稳定性，现代自然语言处理模型能够有效地处理长文档，同时优化性能和资源利用（Al-Qurishi [2022] ）。

与基于模型的方法不同，基于数据的方法着重于修改输入到模型的训练和测试数据。这些方法旨在通过优化数据管道来提高模型的性能和效率，而不是改变模型本身。一个显著的以数据为中心的技术是判别式主动学习（DAL），这是一种高级策略，旨在最大限度地减少标注工作量，同时最大化模型性能。DAL 通过识别位于模型决策边界附近的数据点来选择最有信息量的实例进行标注，从而使标注数据的分布在学习的表示空间中与未标注池中的数据不可区分。这种选择性采样过程减少了达到高准确性所需的标注示例数量。另一种有效的基于数据的策略解决了基于 Transformer 模型的输入长度限制。一个常见的方法是将长文档拆分为较小的、可管理的段，这些段分别处理后进行聚合以进行最终分类。在这种情况下，经常使用分层模型，首先处理来自文档块的局部信息，然后由更高级别的模型组合输出。分层注意机制通过选择性地关注文档中最相关的部分，进一步提高了效率，从而避免了需要一次性编码整个文档。在我们的工作中，我们优先采用以数据为中心的方法，因为它可以无缝集成到现有模型中、适用于不同领域、不容易受到模型所引发的偏见的影响，并且可以扩展到各种任务中（Song [2024], Moro [2023] ）。具体来说，我们专注于在训练和推理过程中尽

量减少提供给模型的上下文信息，从而提高计算效率（He [2019], Liu et al. [2018b], Tay et al. [2021] ）。这是通过选择性地策划和简化输入数据来实现的，而不是改变模型的结构。这样的方法能够在不涉及结构性修改的情况下，提高在各种领域和模型类型中的适应性（Li et al. [2018], Prabhu et al. [2021], Sun et al. [2020] ）。

3 方法论

我们实验中使用的数据集来源于 L3Cube 的 IndicNews 语料库，这是一个为印度地区语言设计的多语言文本分类数据集。MahaNews 对应于 IndicNews 数据集的马拉地语子集。该语料库涵盖了 11 种主要的印度语言的新闻标题和文章，每种语言数据集包含 10 个或更多的新闻类别。我们使用了长文档分类（LDC）数据集，该数据集由完整的文章及其类别组成（Mittal et al. [2023], Aishwarya et al. [2023] ）。

我们的方法学集中于在使用马拉地 LDC 数据集的过程中优化输入尺寸，同时保持分类性能，马拉地 LDC 数据集由马拉地语言的完整文章组成（Jain et al. [2020] ）。我们首先将每篇文章分词成单个句子，然后计算每个句子的 TF-IDF 分数。接着根据分数对句子进行排序，并通过选择排名靠前的句子来减少上下文。为了实现这一点，我们探索了各种句子选择策略。在不使用整篇文章的情况下，选出的句子被输入到 MahaBERT 模型中进行分类。

3.1 训练和测试

我们的目标是通过减少训练和推理过程中输入文本的数量来提高分类效率，同时保持与全文档设置相当的性能。当在完整的 LDC 数据集上进行训练和评估时，在 L3Cube-MahaCorpus 和其他公开的马拉地语数据集上微调的 MahaBERT 模型取得了 94.706% 的准确率。我们的目标是使用减少上下文的输入来接近这个准确率，从而降低计算成本和训练及测试时间。马拉地语 LDC 数据集包括 20425 个训练样本和 2550 个测试样本。此外，数据集中还有 2548 个验证样本，有助于提高已训练模型的准确性。

句子选择技术

为实现减少上下文的分类，我们评估了几种句子选择策略，从简单的选择方法到一种新颖的基于 TF-IDF 的方法。这些方法旨在保留每个文档中最具信息量的部分，用于训练模型并随后对其进行测试。

尽管这些方法简单易行，但它们可能无法始终如一地捕捉到文档中最相关的内容，因为重要的句子可能出现在文本的各个部分。

3.2 基于 TF-IDF 的排序和选择

尽管随机选择句子在计算上是高效的，但没有考虑到句子在文档中的相对重要性，这取决于独特性和语义相关性等因素。这导致分类准确性不一致且不可靠，特别是在长文档分类任务中。

为了解决这些限制，我们提出了一种基于 TF-IDF 分数的新颖句子选择技术，它根据句子的信息价值进行排序。该方法在显著减少推理和训练时间的同时保持高分类精度，为长文档分类提供了一个有效且可扩展的解决方案。

方法的总体流程

每篇文章的句子按照原始顺序保留，并使用特定语言的分词器进行分词，例如 IndicNews 数据集中的 Indic NLP 分词器。

为了计算 TF-IDF 分数，每个句子在确定词频 (TF) 和逆文档频率 (IDF) 的上下文中被视为独立的文档。这种方法识别出在句子中频繁出现但在同一文章中其他句子中较少出现的词，从而量化每个词的重要性。

在为文章中的所有不同词计算 TF-IDF 分数后，每个句子都会收到一个累积的 TF-IDF 分数，这个分数是从其组成词汇中聚合而来的。句子然后根据这些累积分数进行排序，结果是一个有序数组，其中最具信息的句子出现在顶部。

分数计算

一个句子 Si 的分数可以被计算为句子中所有词语 t_j 的 TF-IDF 分数之和。

形式上，分数分数 (S_i) 被定义为：

$$\text{Score}(S_i) = \sum_{t_j \in S_i} \text{TF-IDF}(t_i)$$

其中，

$$\text{TF-IDF}(t_i) = \text{TF}(t_i) \cdot \text{IDF}(t_i)$$

3.3

定义

1. 词频 (TF)：

$$\text{TF}(t_j) = \frac{\text{Frequency of } t_j \text{ in } S_i}{\text{Total number of terms in } S_i}$$

2. 逆向文档频率 (IDF):

$$\text{IDF}(t_j) = \log\left(\frac{N}{1 + \text{Sentence frequency of } t_j}\right)$$

其中 N 是文章中句子的总数，文档频率是包含 tj 的句子的数目。
这个公式确保每个句子的重要性来源于其术语在文章语境中的意义。
上述选择句子的方式用于模型训练，如图 2 所示。

最优句子选择数量：

选择最优的句子数量需要在训练效率和分类准确性之间进行权衡。考虑了几种方法来确定最具信息量的句子子集：

这些方法有助于优化句子选择策略，以提高计算效率和模型性能，同时将不必要的信息降到最低。

归一化是一种数据预处理技术，通过调整特征或变量的值到一个共用的尺度，以不扭曲值的范围差异。这将分数调整到一个特定的数值范围内，并确保所有特征都是在同一个尺度上，没有单个特征因为其尺度而占主导地位，从而允许公平地贡献于计算。在我们的情况下，归一化是必要的，因为我们根据每个句子的 TF-IDF 分数总和对其进行评分，并进行排名以选择用于训练和测试的顶级句子。如果没有归一化，更长的句子（包含更多词）将会自然地拥有更高的 TF-IDF 总和，仅仅因为它们包含更多的词，而不是因为其中包含更多重要的词。这将产生一个偏差，使得更长的句子排名更高，即使它们没有成比例地更多信息性内容。为了确保公平的句子排名，不同的方法平衡句子长度和 TF-IDF 分数：通过使用句子长度进行归一化（即，将总 TF-IDF 分数除以句子中的词数），我们确保排名反映句子中词的平均重要性，而不是绝对总和。这有助于公平地比较不同长度的句子，防止偏向更长的句子，并确保选择基于词的重要性而不是句子长度。然而，在分析所选择的句子后，观察到归一化引入了一个逆向偏差，使得算法更倾向于较短的句子。解决此问题的替代方法是引入一个附加因素，以确保一个更平衡且有意义的句子评分过程。平衡长度因子

为了实现公平排序，我们需要一种机制能够动态调整 TF-IDF 得分和句子长度的影响。这种方法创建了一种灵活的排序机制，其中可以控制每个因素的相对重要性，以确保在独特性和上下文之间的最佳权衡。
为了平衡这种偏倚并在两个极端之间实现权衡，引入了以下公式：

$$\text{Score} = (\lambda_1 \cdot \text{Normalized\_TF\_IDF}) + (\lambda_2 \cdot \text{length})$$

其中，$\lambda_1 = 1 - \lambda_2 \quad \text{and} \quad 0 \leq \lambda_1, \lambda_2 \leq 1$

这些是用于控制 Normalized_TF_IDF 和长度在最终排序中相对重要性的权重。

- $\lambda_1 > \lambda_2$：关注包含独特词汇（较高 TF-IDF 分数）的句子。
- $\lambda_2 > \lambda_1$：优先考虑包含更多上下文的句子（较长的句子）。

这个公式通过在两个因素之间分配总权重，有效地平衡了由于归一化和句子长度引入的偏差。由于 $\lambda_1 = 1 - \lambda_2$，增加一个因素的权重会自动减少另一个因素的影响，从而确保受控的权衡。如果 λ_1 较高，排序就偏向于 TF-IDF 分数更高的句子，注重术语的独特性。相反，如果 λ_2 较高，则更长的句子更具上下文丰富性，将被优先考虑。这个动态的加权机制允许根据分类任务的特定需要进行微调，以防止对于较短或较长句子的极端偏向。

Sentence(S)	First	Last	Random	Ranked
1	90.70 %	81.64 %	75.76 %	90.35 %
2	93.01 %	87.60 %	90.31 %	93.17 %
3	93.17 %	89.76 %	91.49 %	93.64 %
4	92.82 %	91.09 %	91.72 %	94.00 %
5	93.56 %	91.64 %	92.70 %	94.19 %

Table 1: 按句准确率结果——表格显示了不同句子选择策略（首句、末句、随机、排序）在选择 1 到 5 个句子时的准确率。结果表明，排序方法表现最佳，其次是首句、随机和末句，这突出了选择方法和句子数量对模型性能的重要性。

4 结果与讨论

4.1 句子数量法

如表 1 和图 3 所示，结果展示了对应于选择每个文档的第一个、最后一个、随机的、以及排名前 1、2、3、4 和 5 个句子的准确性。一种明显的趋势是，随着所选句子数量的增加，准确性提高，这反映了包括更多内容的额外信息价值。这种提升在第一个、最后一个和随机选择方法中一致观察到，尽管在排名方法中在三个句子时达到峰值。

通过选择固定数量的排名靠前的句子，而不考虑文档长度，推理时间显著减少，并在不同文档中几乎保持不变。这种方法仅用 5 个句子就实现了 94.19 % 的准确率，相较于完整上下文基准准确率 94.706 %，仅下降了 0.544 %。这表明可以在对准确率影响极小的情况下，实现输入长度的显著减少。

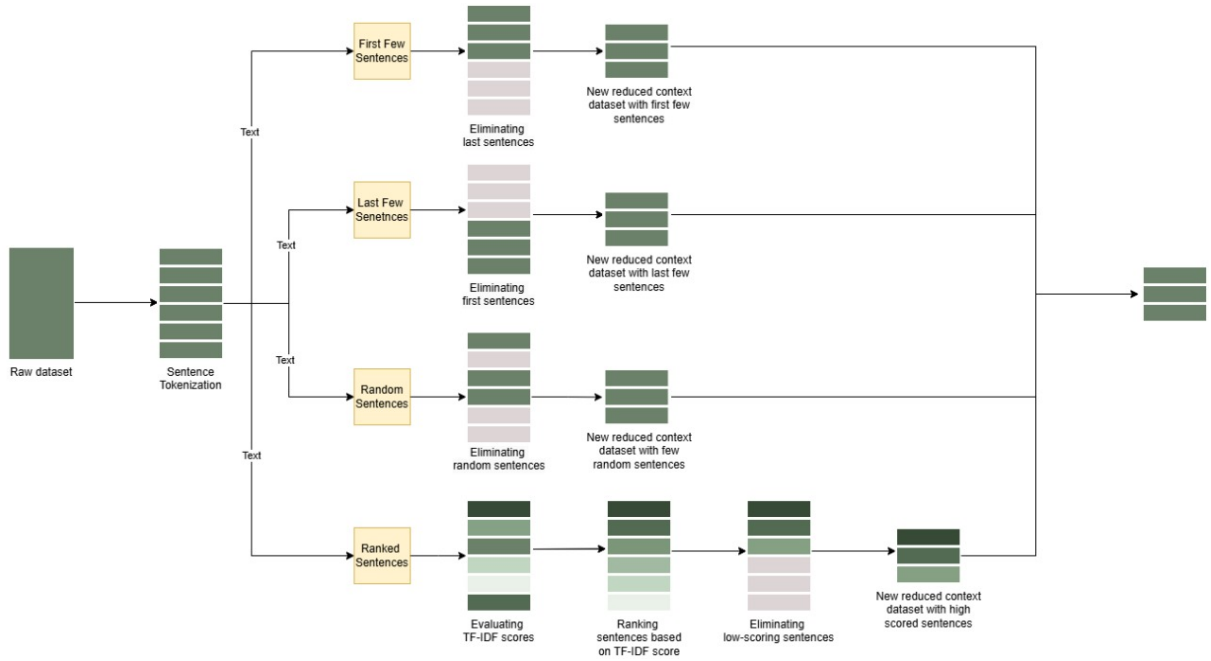

Figure 2: 句子选择方法---图像展示了用于上下文缩减的各种句子选择方法。方法包括选择前几句、后几句、随机句子和排序句子。在排序的方法中，句子使用 TF-IDF 进行评分并根据相关性进行选择。这些策略有助于在保留有意义的上下文的同时减少输入大小。

Sentence(S)	0.2 (λ_2)	0.5 (λ_2)	0.7 (λ_2)	1.0 (λ_2)
1	89.11 %	88.94 %	89.10 %	89.26 %
2	92.82 %	91.86 %	92.73 %	92.23 %
3	93.79 %	93.81 %	93.78 %	93.82 %
4	93.95 %	93.36 %	94.07 %	93.59 %
5	93.47 %	93.56 %	93.67 %	93.32 %

Table 2: 归一化的句子准确率结果 — 该表格展示了用于排序句子选择方法的归一化句子准确率结果。它显示了 λ_2 从 0.2 到 1.0 的不同值的结果。$\lambda_2 = 0.7$ 为不同的句子数量提供了最佳性能。

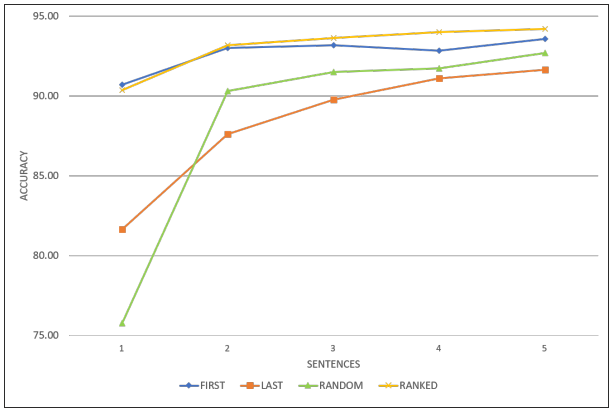

Figure 3: 按句准确度图---该图展示了不同选择方法的按句准确度：首次选择、最后选择、随机选择和排序选择。图中将准确度与选定句子数量（1 到 5）作了对比。图表显示排序选择 > 首次选择 > 随机选择 > 最后选择。

虽然固定长度的顶级句子选择能产生强有力的结果，但它将所有高得分的句子一视同仁，而不考虑句子的长度或信息比例。为了进一步完善这种方法，我们引入了一种标准化策略，将标准化的 TF-IDF 得分与句子长度相结合，以更好地捕捉相关性和信息内容。

正规化结果

表格 2 显示了在 1 到 5 个选定句子中，将标准化的 TF-IDF 分数与句子长度结合使用各种权重 ($\lambda_2 = 0.2, 0.5, 0.7, 1.0$) 的影响。准确率在 $\lambda_2 = 0.7$ 时达到峰值，表明在最小上下文中，较长的句子更具信息性。随着更多句子被纳入，TF-IDF 的重要性增加。最佳结果出现在 $\lambda_2 = 0.2$ 和 0.7 时，平衡了相关性和长度。最高准确率为 94.07 %，在 $\lambda_2 = 0.7$ 时使用 4 个句子实现。随着句子数量的增加，准确率趋于稳定，表明随着模型对权重变化的敏感性下降，从进

Sentence(S)	Ranked	Ranked Normalized
1	90.35 %	89.11 %
2	93.17 %	92.82 %
3	93.64 %	93.82 %
4	94.00 %	94.07 %
5	94.19 %	93.67 %

Table 3: 排序和排序归一化结果的比较—该表比较了使用排序和排序归一化方法进行句子选择的准确性。结果表明，当选定句子的数量较少时，归一化几乎没有影响。

一步调整中获得的回报递减。

表 3 比较了基础排序句子选择方法与标准化排序策略的分类性能，其中标准化的 TF-IDF 分数与句子长度按每个情况使用最佳加权参数（λ_1 和 λ_2 ）相结合。结果显示标准化没有显著改进，表明当仅选择少数句子时，这种方法的影响最小。表 **??** 和图 **??** 展示了从文档中选择第一、最后、随机和按比例排序的句子所达到的准确度。使用完整的文档进行训练和测试可获得 94.706 % 的准确度，作为比较的基准。重要的是，通过将上下文减少到原始文档的 40% 到 50%，我们仍然能实现 94.39 % 的惊人准确度，与基准准确度 94.706 % 的差距微小，表明其性能与使用完整文档上下文的方法相当。随着上下文大小的减少，排序选择方法始终优于其他技术，如第一、最后和随机选择。随着上下文大小的增加，所有方法的性能趋于一致，结果类似。这种收敛表明，排序选择方法在较小的上下文窗口中特别有效地提高了准确性。在这种情况下，标准化显示出积极影响，在大多数情况下提高了性能。

4.2 推理时间

语境缩减的目标是在不牺牲分类准确性的情况下最小化推理时间。我们的方法使用 TF-IDF 有效地削减语境，使序列长度与每个文档的实际内容对齐，而不是依赖固定的限制（例如，BERT 中的 512 个标记）。使用填充或截断的固定长度会削弱效率的提升，而动态调整则确保了准确和更快的推理。下面图 4 和 5 中的曲线描绘了在选择不同语境时测试时间的变化。
图 4 显示了句子数量与评估时间之间的正相关关系。尽管从 1 到 3 个句子的增长较为温和，但从第 4 个句子开始变得更加明显，表明评估时间随着句子数量的增加而增长越来越明显。
图 5 描绘了评估的内容百分比与相应评估时间之间的关系。从 10 % 到 20 %，评估时间保持相对稳定，然后从 30 % 开始急剧增加，在 100 % 达到峰值。这表明了强烈的正相关性——而

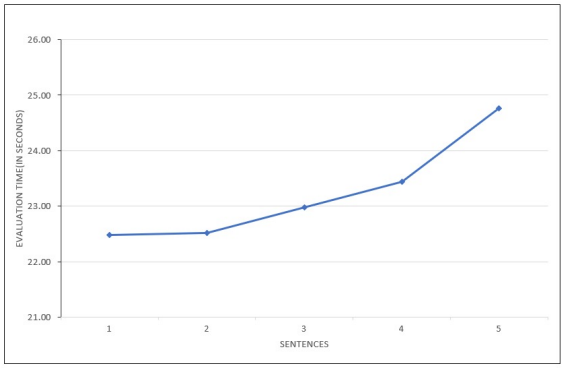

Figure 4: 句子逐一选择的评估时间图——该图展示了在句子逐一选择过程中，评估时间（以秒为单位）与所考虑的句子数量之间的关系。x 轴表示句子数量，而 y 轴则显示相应所需的评估时间。该图通常突出了一个趋势，即随着句子数量的增加，评估时间也增加。

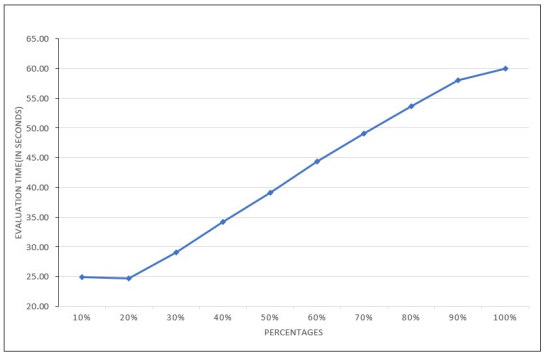

Figure 5: 按百分比选择的评估时间图 — 该图展示了评估时间（以秒为单位）与在百分比选择过程中被选句子百分比之间的关系。图中演示了随着所选百分比的变化，评估时间也随之变化的趋势。

且是越来越非线性的，尤其是在极端情况下，这可能是由于更高的计算负荷。

值得注意的是，在仅使用原始文档上下文的 40 % 时，我们的方法实现了 94.39 % 的准确性，仅比全上下文基线低 0.33 %，同时将推理延迟减少了 43 %。这表明在速度和性能之间实现了有效的权衡，使该方法非常适合于既需要准确性又需要可扩展性的实际应用。

5 结论

我们引入了一种利用句子选择技术进行长文档分类的高效方法，该方法在减少输入大小的同时保持了与全上下文模型相当的准确性。传统的基于 transformer 的模型由于输入长度限制和计算成本，在处理长文本时存在困难。我们的方法通过首/尾句选择、随机抽样和基于 TF-IDF 的排序等策略解决了这些限制。

在 L3Cube 的 IndicNews 集合中的 Marathi 长文档分类（LDC）数据集上进行的实验表明，我们的方法在不影响性能的情况下显著降低了计算成本。基于 TF-IDF 的排序在识别用于分类的信息句子方面被证明特别有效。

我们还探讨了输入大小和准确性之间的权衡，证明选择一部分高排名句子而不是一个固定数量可以在效率和性能之间取得更好的平衡。将正规化引入排名即使在减少输入大小的情况下也能进一步提高准确性。

总之，我们的方法为长文档分类提供了一种可扩展且资源高效的解决方案。未来的工作可能涉及领域特定的句子选择或混合模型，以进一步优化输入表示并提升在不同自然语言处理任务中的性能。

该工作是在 Pune 的 L3Cube 的指导下进行的。我们真诚地感谢导师在此过程中提供的宝贵指导和持续鼓励。

References

Mirashi Aishwarya, Sonavane Srushti, Lingayat Purva, Padhiyar Tejas, and Joshi Raviraj. L3cube-indicnews: News-based short text and long document classification datasets in indic languages. In *Proceedings of the 20th International Conference on Natural Language Processing (ICON)*, pages 442–449, 2023.

Muhammad Al-Qurishi. Recent advances in long documents classification using deep-learning. In *Proceedings of the 5th International Conference on Natural Language and Speech Processing (ICNLSP 2022)*, 2022. doi: 10.18653/v1/2022.icnlsp-1.12.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020.

Bijoyan Das and Sarit Chakraborty. An improved text sentiment classification model using tf-idf and next word negation, 2020.

Mamata Das, K. Selvakumar, and P.J.A. Alphonse. A comparative study on tf-idf feature weighting method and its analysis using unstructured dataset, 2023.

J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

Jun He. Long document classification from local word glimpses via recurrent attention learning. *IEEE Access*, 2019. doi: 10.1109/ACCESS.2019.2907992.

Kushal Jain, Adwait Deshpande, Kumar Shridhar, et al. Indic-transformers: An analysis of transformer language models for indian languages, 2020.

Raviraj Joshi. L3cube-mahacorpus and mahabert: Marathi monolingual corpus, marathi bert language models, and resources. In *Proceedings of the WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*, pages 97–101, 2022.

Sang-Woon Kim and Joon-Min Gil. Research paper classification systems based on tf-idf and lda schemes. *Human-centric Computing and Information Sciences*, 2019. doi: 10.1186/s13673-019-0192-7.

Chao Li, Yanfen Cheng, and Hongxia Wang. A novel document classification algorithm based on statistical features and attention mechanism, 2018.

C. z. Liu, Y. x. Sheng, Z. q. Wei, and Y. Q. Yang. Research of text classification based on improved tfidf algorithm. In *2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE)*, 2018a. doi: 10.1109/IRCE.2018.8492945.

L. Liu, K. Liu, Z. Cong, J. Zhao, Y. Ji, and J. He. Long length document classification by local convolutional feature aggregation. *Algorithms*, 2018b. doi: 10.3390/a11080109.

S. Minaee, N. Kalchbrenner, E. Cambria, et al. Deep learning based text classification: A comprehensive review, 2021.

Saloni Mittal, Vidula Magdum, Sharayu Hiwarkhedkar, Omkar Dhekane, and Raviraj Joshi. L3cube-mahanews: News-based short text and long document classification datasets in marathi. In *International Conference on Speech and Language Technologies for Low-resource Languages*, pages 52–63. Springer, 2023.

G. Moro. Efficient memory-enhanced transformer for long-document summarization in low-resource regimes. *Sensors*, 2023. doi: 10.3390/s23073542.

Hyunji Park, Yogarshi Vyas, and Kashif Shah. Efficient classification of long documents using transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2022. doi: 10.18653/v1/2022.acl-short.79.

Sumanth Prabhu, Moosa Mohamed, and Hemant Misra. Multi-class text classification using bert-based active learning, 2021.

Shahzad Qaiser and Ramsha Ali. Text mining: Use of tf-idf to examine the relevance of words to documents. *IJCA*, 2018. doi: 10.5120/ijca2018917395.

B. Song. State space models based efficient long documents classification. *Journal of Intelligent Learning Systems and Applications*, 2024. doi: 10.4236/jilsa.2024.163009.

Chi Sun, Xipeng Qiu, Yige Xu, et al. How to fine-tune bert for text classification?, 2020.

Y. Tay, M. Dehghani, S. Abnar, et al. Long range arena: A benchmark for efficient transformers, 2021.

Vedangi Wagh, Snehal Khandve, Isha Joshi, Apurva Wani, Geetanjali Kale, and Raviraj Joshi. Comparative study of long document classification. In *TENCON 2021-2021 IEEE Region 10 Conference (TENCON)*, pages 732–737. IEEE, 2021.

M. Zaheer, J. Ainslie, G. Guruganesh, et al. Big bird: Transformers for longer sequences. In *Proceedings of NeurIPS*, pages 702–709, 2020. doi: 10.48550/arXiv.2007.14062.