Graph-tree Fusion Model with Bi-Directional Information Propagation for Long Document Classification

Anonymous ACL submission

Abstract

Long document classification presents challenges in capturing both local and global dependencies due to their extensive content and complex structure. Existing methods often struggle with token limits and fail to adequately model hierarchical relationships within documents. To address these constraints, we propose a novel model leveraging a graph-tree structure. Our approach integrates syntax trees for sentence encodings and document graphs for document encodings, which capture fine-grained syntactic relationships and broader document contexts. We use Tree Transformers to generate sentence encodings, while a graph attention network models inter- and intra-sentence dependencies. During training, we implement bidirectional information propagation from word-to-sentence-to-document and vice versa, which enriches the contextual representation. Our proposed method enables a comprehensive understanding of content at all hierarchical levels and effectively handles arbitrarily long contexts without token limit constraints. Experimental results demonstrate the effectiveness of our approach in all types of long document classification tasks.

1 Introduction

Long document understanding has garnered increasing attention in the field of natural language processing (NLP) due to its wide range of applications across various domains, including legal document analysis, scientific literature categorization, and clinical text mining. Accurate understanding of long documents is essential for tasks such as information retrieval, content summarization, and decision-making support systems. Modern deep learning models for semantic analysis achieve impressive results by training on large datasets, which enables them to generate highly accurate predictions on unseen content (Al-Qurishi, 2022). However, their ability to capture relationships between words and sentences relies on increasingly complex statistical operations as the text sequence lengthens (Tay et al., 2020). Consequently, many existing methods become impractical for real-world applications, which makes processing long documents a challenging task.

One of the primary challenges of long document classification is managing the large volume of information. Unlike short texts, long documents contain extensive content that often spans multiple topics, making it difficult to capture the overall context of the document effectively. Transformer-based models (Vaswani et al., 2017), such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), GPT-3 (Brown et al., 2020) and LLaMa-2 (Touvron et al., 2023), had gained popularity for NLP tasks due to their ability to capture (relatively) long-range dependencies and contextual relationships. However, in long document classification, transformer models face scalability issues due to their quadratic time complexity with respect to the input length. Processing long documents with transformers can be computationally expensive and memory-intensive, often requiring substantial hardware resources. Current methods for handling lengthy documents include truncating texts to a predefined length or modifying the attention mechanism. Truncating to the first 512 tokens is straightforward but may cause significant information loss. Sparse attention models like Longformer (Beltagy et al., 2020) and Big Bird (Zaheer et al., 2020) reduce computational load by focusing on a subset of tokens, but they do not fully capture comprehensive context and dependencies in long texts.

Another significant challenge in long document classification is capturing contextual dependencies and the hierarchical structure. Long documents have dependencies at word, sentence, and document levels, which are crucial for understanding content and leveraging both local and global contexts. This structure is important for understand-

ing the overall context and meaning. Treating text as a flat sequence of tokens can cause models to miss these important hierarchical relationships. Current approaches to address these challenges involve hierarchical models. For instance, Hierarchical Attention Network (HAN) (Yang et al., 2016) and Hierarchical Attention Transformers (HAT) (Chalkidis et al., 2022), aim to capture both word-level and sentence-level representations before aggregating them into document-level embeddings. However, these models often fail to capture the intricate relationships between different parts of the document, such as the interplay between words, sentences, and overall document structure (Dai et al., 2019). Additionally, hierarchical models may struggle with long-range dependencies (Dong et al., 2023), which misses the relationships between distant sections essential for understanding the overall context.

To address the aforementioned challenges and constraints, we propose a novel model that leverages a graph-tree structure for arbitrarily long document classification. Our approach fuses syntax trees for sentence encodings with document graphs for document encodings, which provides a comprehensive representation that captures both local and global dependencies. Syntax trees (both dependency and constituency) represent the grammatical structure of sentences, which enhances sentence-level understanding. We use Tree Transformers (Ahmed et al., 2019a) to generate sentence encodings from these syntax trees. The document graph preserves hierarchical relationships within the document, which ensures that both local and global contexts are considered during the classification process. We apply the Graph Attention Network (GAT) (Veličković et al., 2018) on the constructed document graph to model the dependencies between the sentences. This graph structure effectively captures both inter- and intra-sentence dependencies. Meanwhile, during training, we implement a bidirectional information propagation approach where information flows both from word-to-sentence-to-document and from document-to-sentence-to-word. This bidirectional flow enriches the contextual representation of the document, which allows for a more comprehensive understanding of the content at all hierarchical levels. By incorporating syntax trees and document graphs, we also encode the semantic units differently according to their characteristics. This allows our model to handle arbitrarily long contexts without being constrained by token limits.

To summarize, our main contributions are:

1. We introduce a novel graph-tree structure that combines syntax trees and document graphs to capture both local and global dependencies within arbitrarily long documents.
2. We introduce a bidirectional information propagation approach where the information flows both from word-to-sentence-to-document and from document-to-sentence-to-word, which enriches the contextual representation of the document.
3. We show empirically that our model achieves improvements across a variety of classification tasks, including binary, multi-class, and multi-label classification.

2 Preliminaries

2.1 Tree Transformer

Tree Transformer (Ahmed et al., 2019a) is designed to more effectively preserve syntactic and semantic information. Given a dependency or constituency tree structure of a sentence, a dependency tree has a word at every node, represented by $\mathbf{X}_d$ while, in a constituency tree, only the leaf nodes contain words, represented by $\mathbf{X}_c$:

$$X_d = \begin{bmatrix} \mathbf{p}_v \\ \mathbf{c}_1^v \\ \mathbf{c}_2^v \\ \vdots \\ \mathbf{c}_n^v \end{bmatrix}, \quad X_c = \begin{bmatrix} \mathbf{c}_1^v \\ \mathbf{c}_2^v \\ \vdots \\ \mathbf{c}_n^v \end{bmatrix}, \tag{1}$$

where $\mathbf{p}_v$ is the initial parent representation and $\mathbf{c}_i^v$ is the initial child representation of node i.

The parent node embedding $\mathbf{P}$ is computed using multi-branch attention built upon multi-head attention in the vanilla Transformer (Vaswani et al., 2017). The branch attention $\mathbf{B}_i$ for branch i is computed as :

$$\mathbf{B}_i = \text{Attention}(\mathbf{Q}_i\mathbf{W}_i^Q, \mathbf{K}_i\mathbf{W}_i^K, \mathbf{V}_i\mathbf{W}_i^V), \tag{2}$$

where $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, and $\mathbf{W}_i^V$ are the learnable weight matrices. Each $\mathbf{B}_i$ is then normalized and scaled using a layer normalization block (Ba et al., 2016):

$$\overline{\mathbf{B}}_i = \text{LayerNorm}(\mathbf{B}_i\mathbf{W}_i^b + \mathbf{B}_i) \times \kappa_i, \tag{3}$$

where $\mathbf{W}_i^b$ and κ_i are the learnable parameters. Then, a position-wise convolutional neural network

(PCNN) is applied to each $\overline{\mathbf{B}}_i$, and the branch attention is aggregated:

$$\text{BranchAttn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \sum_{i=1}^{n} \alpha_i \text{PCNN}(\overline{\mathbf{B}}_i), \tag{4}$$

where α_i is learnable. The final parent representation, or sentence embedding, is obtained by:

$$\mathbf{P}' = \text{EwS}(\tanh((\mathbf{x}' + \mathbf{x})\mathbf{W} + b)), \tag{5}$$

where EwS is element-wise summation, and $\mathbf{x}$ and $\mathbf{x}'$ depicts the input and output of the attention module. Additional details on the full operations of the Tree Transformer are in the original paper (Ahmed et al., 2019a).

2.2 Graph Attention Network

Graph Attention Network (GAT) (Veličković et al., 2018) is designed to model information flow between nodes, which enhances node representations by employing attention over features from neighbouring nodes. Given a heterogeneous graph $G = (V, E)$ with N nodes, the input node features $h = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N\}$, and GAT layer with multi-head attention is designed as follows:

$$\begin{aligned} e_{ij} &= \text{LeakyReLU}\left(\mathbf{a}^T \left[\mathbf{W}\mathbf{h}_i \parallel \mathbf{W}\mathbf{h}_j\right]\right), \\ \alpha_{ij} &= \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}, \\ \mathbf{h}'_i &= \Big\|_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \mathbf{h}_j\right), \end{aligned} \tag{6}$$

where $\|$ denotes concatenation, $\mathbf{W}$ is a weight matrix, $\mathbf{a}$ is a weight vector, $\mathcal{N}_i$ is the neighbourhood of node i, LeakyReLU and σ are the activation functions, K is the number of attention heads, α_{ij}^k and $\mathbf{W}^k$ are the attention coefficients and weight matrix for the k^{th} head, respectively.

3 Method

We propose a novel Graph-Tree Fusion Model, as shown in Figure 1 that leverages a graph-tree structure for long document classification. Our model uses multi-granularity document representations through Tree Transformers and graph attention networks. It fuses syntax trees for sentence encodings with document graphs for document encodings. Additionally, during training, we implement a bidirectional information propagation approach, allowing information to flow both from word-to-sentence-to-document and from document-to-sentence-to-word.

3.1 Sentence Encoder via Syntax Tree

Sentences are foundational units of a document. Preserving sentence semantics enhances the accuracy and informativeness of document embeddings.

We first split a document $\mathcal{D}$ into sentences as $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$. Given a sentence s with a sequence of input tokens, we use RoBERTa (Liu et al., 2019) to encode the tokens and output the corresponding vector for each token from the last hidden layer, denoted as $\mathbf{E}(s) = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n\}$, where $\mathbf{e}_i$ is the embedding for token i. We then parse the sentences, obtaining initialized sentence embeddings $\mathbf{X}_d^{s_i}$ for the dependency tree and $\mathbf{X}_c^{s_i}$ for the constituency tree (see equation 1). These two tree structures are then processed by the Tree Transformers mentioned in Section 2.1, yielding enhanced sentence representations for each sentence: $\mathbf{h}_d^{s_i}$ for the dependency tree and $\mathbf{h}_c^{s_i}$ for the constituency tree:

$$\begin{aligned} \mathbf{h}_d^{s_i} &= \text{TreeTransformer}(\mathbf{X}_d^{s_i}), \\ \mathbf{h}_c^{s_i} &= \text{TreeTransformer}(\mathbf{X}_c^{s_i}). \end{aligned} \tag{7}$$

The final embedding for sentence i is defined as the mean-pooling of the two tree representations:

$$\mathbf{h}^{s_i} = \frac{1}{2}(\mathbf{h}_d^{s_i} + \mathbf{h}_c^{s_i}). \tag{8}$$

3.2 Document Encoder via Document Graph

Sentence Selector Using Label-wise Attention. To identify the most important sentences, we calculate the similarity score between each sentence s_i in $\mathcal{D}$ and the labels using label-wise attention. We obtain the label embeddings for each label $\mathcal{Y}_i$ by averaging the word embeddings (from RoBERTa) of each word in their descriptors:

$$\mathbf{h}^{\mathcal{Y}_i} = \frac{1}{m} \sum_{j \in m} w_j, i = 1, 2, \ldots, L, \tag{9}$$

where m is the number of words in the descriptor, and L is the number of labels. The similarity score is:

$$\alpha_{s_i, \mathcal{Y}_i} = \text{Softmax}(\mathbf{h}^{s_i} \cdot \mathbf{h}^{\mathcal{Y}_i}), \tag{10}$$

where $\alpha_{s_i, \mathcal{Y}_i}$ is a probability that contains the scores associated with a sentence i in $\mathcal{D}$ to a specific label $\mathcal{Y}_i$. We then apply a threshold τ to select sentences with high probabilities:

$$\mathcal{S}' = \{s_i | \alpha_{s_i, \mathcal{Y}_i} \geq \tau\}. \tag{11}$$

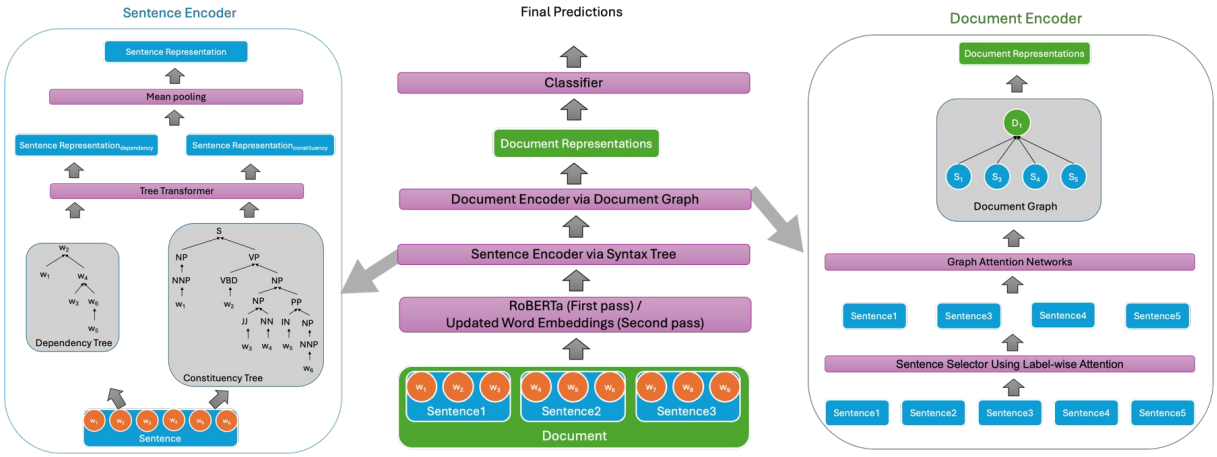

Figure 1: Overview of our proposed model. Our proposed model integrates syntax trees for sentence encodings (showing on the left) and document graphs for document encodings (showing on the right). We employ Tree Transformers to generate sentence encodings from the syntax trees and use graph attention networks to generate document encodings from the document graph. Our model workflow includes two passes. In the first pass, initial word embeddings are obtained from RoBERTa. In the second pass, these word embeddings are updated through bidirectional information propagation.

Document Encoding Using Graph Attention Network (GAT). To obtain the document representation, we construct a heterogeneous document graph $G = (V, E)$ that captures the relations between documents and their sentences. The graph G contains sentence nodes and document nodes, and one type of edge: document-sentence edges. Specially, for each document $\mathcal{D}$, we create a document node $v_{\mathcal{D}}$ and a set of selected sentence nodes $V_{\mathcal{S}'} = \{v_{s_1}, v_{s_2}, ..., v_{s_N}\}$. Directed edges E are established from each sentence node v_{s_i} to its corresponding document node $v_{\mathcal{D}}$.

We apply GAT (described in 2.2) to model the inter-sentence and document relations within a document, which enhances the document representations derived from sentence representations.

To obtain the document representation for $\mathcal{D}$, the sentence node embeddings $\mathbf{h}^{s_i}$ are initially generated using the sentence encoder. The document node feature is then initialized by taking the mean pooling of the features of its sentence nodes:

$$\mathbf{h}^{\mathcal{D}} = \frac{1}{|V_{\mathcal{S}'}|} \sum_{s_i \in V_{\mathcal{S}'}} \mathbf{h}^{s_i}. \tag{12}$$

We use the GAT layer with multi-head attention to compute the new document node features $\mathbf{h'}^{\mathcal{D}}$ as follows:

$$\mathbf{h'}^{\mathcal{D}} = \text{GAT}(\mathbf{h}^{s_i}). \tag{13}$$

After each GAT layer, we introduce a position-wise feed-forward (FFN) layer, consisting of two linear transformations similar to the vanilla Transformer architecture (Vaswani et al., 2017), to obtain the final document representation:

$$\tilde{\mathbf{h}}^{\mathcal{D}} = \text{FFN}(\mathbf{h'}^{\mathcal{D}}). \tag{14}$$

3.3 Bidirectional Information Propagation

Inspired by Wang et al. (2020), we implement a bidirectional information propagation approach, as shown in Figure 2. This approach allows information to flow from word-to-sentence-to-document and from document-to-sentence-to-word.

After obtaining the document representation from the document encoder in Section 2.2, we update the sentence nodes using the updated document nodes and then update the word nodes using the updated sentence encodings. We further iteratively update the document nodes and sentence nodes. The information flow is bidirectional: in each iteration, it first moves from word-to-sentence-to-document (see Sections 3.1,3.2), and then from document-to-sentence-to-word. For the t^{th} iteration, the document-to-sentence-to-word process can be represented as:

$$\begin{aligned}
\mathbf{U}^{t+1}_{\mathcal{D}\to\mathcal{S}} &= \text{GAT}(\mathbf{H}^{t}_{\mathcal{D}}), \\
\mathbf{H}^{t+1}_{\mathcal{S}} &= \text{FFN}(\mathbf{U}^{t+1}_{\mathcal{D}\to\mathcal{S}}), \\
\mathbf{U}^{t+1}_{\mathcal{S}\to w} &= \text{GAT}(\mathbf{H}^{t+1}_{\mathcal{S}}), \\
\mathbf{H}^{t+1}_{w} &= \text{FFN}(\mathbf{U}^{t+1}_{\mathcal{S}\to w})
\end{aligned} \tag{15}$$

where $\mathbf{H}_{\mathcal{D}}$ is initalized by $\tilde{\mathbf{h}}^{\mathcal{D}}$ in Equation 3.2 in the first iteration.

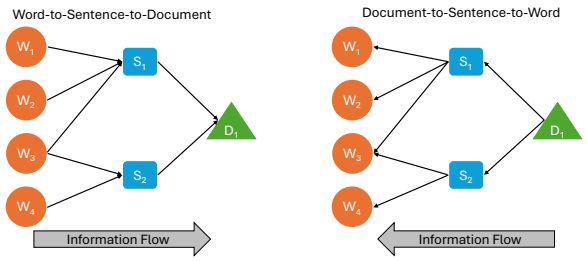

Figure 2: The detailed bidirectional information propagation. Orange, blue, and green nodes represent word, sentence, and document nodes, respectively. The arrows on the edges indicate the current direction of information flow. First, on the left, words are used to aggregate sentence-level information, and the resulting sentence representations are then used to aggregate document-level information. Next, on the right, sentences are updated with the new document representations, and words are updated with the new sentence representations.

As illustrated in Figure 2, word nodes can aggregate document-level information from sentences. For example, a word node with a high degree indicates frequent occurrences in multiple sentences, suggesting it is a keyword of the document. Sentence nodes with a higher concentration of important words are more likely to contain significant information, making them suitable for forming key sections. Bidirectional information propagation enables a comprehensive exchange of information across different hierarchical levels.

3.4 Model Workflow

The architectural structures of the proposed models are portrayed in Figure 1. The proposed model requires two forward passes separated by document-to-sentence-to-word update step. During the initial forward pass, RoBERTa word embeddings are used as the initial input and undergo simultaneous processing by both the Dependency Tree Transformers (DTT) and Constituency Tree Transformers (CTT), and finally mean-pooling (see formula 8) in the sentence encoder. The GAT layer in the document encoder computes the document representation utilizing the sentence representations from the sentence encoder and the first forward pass ends here.

The completion of the first forward pass initiates the activation of the document-to-sentence-to-word update step. The initial step employs the document-to-sentence update step which updates the the $\mathbf{h}_d^{s_i}$ ($\mathbf{h}'^{s_i}_d$)and $\mathbf{h}_c^{s_i}$ ($\mathbf{h}'^{s_i}_c$). Then, the sentence-to-word refinement step is utilized twice: once to update the word embeddings based on the updated $\mathbf{h}'^{s_i}_d$, and another time based on the updated $\mathbf{h}'^{s_i}_c$.

After the document-to-sentence-to-word update step is employed, the second forward pass is initiated. It works over the pruned syntax trees and graph representation of sentences and document nodes given by the first forward pass. This second pass is almost identical to the first pass with a small twist. Here, the sentence encoder works with two different word embeddings. The CTT intakes the word embeddings updated by the $\mathbf{h}'^{s_i}_c$, and the DTT is fed with word embeddings updated by the $\mathbf{h}'^{s_i}_d$ as inputs. The following steps work in the similar manner to the first forward pass. This second forward pass generates refined sentence representations ($\mathbf{h}''^{s_i}_c$, and $\mathbf{h}''^{s_i}_d$) and subsequently the refined document representation ($\mathbf{H}'_{\mathcal{D}}$).

Dataset	Type	# of Classes	# of Instance	Average # of Words per Document
Hyperpartisan	Binary	2	754,000	745
AMZ	Multi-class	5	4850	12,356
20News		20	20,000	369
BOOK	Multi-label	227	16,559	575
ECtHR		33	11,000	5530
Essays		5	1255	660

Table 1: Details of the Long Document Classification Datasets.

4 Experiment

4.1 Setup

Datasets. We evaluate our proposed model on six common long document classification datasets: CMU BOOK Summary (Bamman and Smith, 2013), ECtHR (Chalkidis et al., 2021), Hyperpartisan (Kiesel et al., 2019), 20News (Lang, 1995), Amazon product reviews (AMZ) (He and McAuley, 2016), and Essays(Pennebaker and King, 1999). Following Lu et al. (2023), we randomly sample product reviews longer than 2048 words from the Book category for the AMZ dataset. The statistics of the datasets are summarized in Table 4.1.

Implementation Details. The model employs an initial learning rate of 0.1, which is subsequently reduced by 80% in each iteration if the validation accuracy declines compared to the previous iteration. The batch size is 10. For the tree-transformers, the same hyper-parameter settings are used as in Ahmed et al. (2019b). The statement encoding unit utilizes a GAT with six attention heads. The model's parameters are trained using the "Adagrad" optimizer (Lydia and Francis, 2019). The performance evaluation of our models has been conducted using 10-fold cross-validation. To facili-

tate this cross-validation process, we have utilized the StratifiedKFold function from the scikit-learn package. All experiments have been conducted in an Ubuntu 22.04 LTE environment, leveraging a 48GB NVIDIA RTX A6000 GPU. For parsing the sentences and generating the tree representations, we have used the Stanford Core-NLP parser (Manning et al., 2014).

4.2 Baseline Models

Following Lu et al. (2023), we compare our methods with Transformer-based model with pre-training and State-Space Model (SSM) system.

BERT with Pre-training This simplest approach involves fine-tuning BERT (Devlin et al., 2019) after truncating long documents to the first 512 tokens. A fully connected layer is then applied to the [CLS] token for classification.

ToBERT Transformer over BERT (ToBERT) is a hierarchical approach designed to process documents of any lengths (Pappagari et al., 2019). It divides long documents into chunks of 200 tokens and applies a Transformer layer to the BERT-based representations of these chunks.

Longformer is designed to handle longer input sequences with efficient self-attention that scales linearly with the sequence length, allowing it to process up to 4,096 tokens (Beltagy et al., 2020).

BERT+TextRank To address BERT's 512-token limitation, Park et al. (2022) augment the first 512 tokens with a second set of 512 tokens selected using TextRank (Mihalcea and Tarau, 2004), an efficient unsupervised sentence ranking algorithm.

BERT+Random As an alternative method to BERT+TextRank, Park et al. (2022) augment the first 512 tokens by selecting random sentences up to an additional 512 tokens.

Hungry Hungry Hippo with Max Pooling (H3-pooler) H3 (Fu et al., 2023) is an SSM-based method designed for simultaneous multi-object tracking, maintaining and updating object states based on observed data. Lu et al. (2023) enhanced this model by inserting a max pooling layer between each SSM block, creating the H3-Pooler.

5 Results and Discussions

5.1 Performance Comparision

We compare our proposed model against previous baseline models on various evaluation metrics, as shown in Table 2. We report accuracy for binary and multi-class tasks (Hyperpartisan, 20News, and AMZ) and macro-F1 scores for the multi-label classification problems (BOOK, ECtHR, and Essays). Each row in the table presents the performance of a specific method on each dataset, with the best score for each dataset highlighted.

The results demonstrate that our model consistently achieves superior performance across all datasets. For binary classification (Hyperpartisan), our model outperforms existing methods by approximately 1.3%. In multi-class classification tasks (20News and AMZ), our model shows a marked improvement, exceeding baseline accuracies by about 2% to 4%. In multi-label classification tasks (BOOK, ECtHR, and Essays), our model demonstrates a significant enhancement, with macro-F1 scores improving by approximately 3% to 10% over the best baseline models. These results underscore the robustness and effectiveness of our proposed model in handling the complexities of long document classification across different types of classification tasks.

5.2 Ablation Studies

We are interested in studying the effectiveness and robustness of our model by analyzing various components, such as the tree structure, graph structure, and bidirectional information propagation. To understand the impacts of these factors, we conduct controlled experiments with three different settings: (a) removing the tree structure, which is further divided into three sub-experiments: removing the CTT structure, removing the DTT structure, and replacing the entire tree structure with the [CLS] token; (b) replacing the GAT with a max-pooling layer; and (c) removing the bidirectional information propagation. This allows us to evaluate the influence of each module individually, without interference from the others. The results are summarized in Table 3.

Effectiveness of the Tree Structure. Table 3 demonstrates the crucial role of the tree structure in our model's performance. Removing the CTT component notably decreases performance, especially on multi-class classification datasets like 20News and AMZ, indicating CTT's importance in capturing document context at the phrase level. Similarly, excluding the DTT component leads to the reduction of the performance, with scores dropping by 4-6% across various datasets, which underscores DTT's role in capturing inter-word relations while

Models	Hyperpartisan	20News	AMZ	BOOK	ECtHR	Essays
BERT w/ pre-training (Devlin et al., 2019)	91.8	84.7	51.1	58.2	71.7	69.31
ToBERT (Pappagari et al., 2019)	89.5	85.5	54.6	57.3	77.2	72.2
Longformer (Beltagy et al., 2020)	93.7	83.4	56.4	58.5	81.5	74.38
BERT+Random (Park et al., 2022)	89.3	85.0	56.8	59.2	72.8	70.1
BERT+TextRank (Park et al., 2022)	91.2	84.7	56.9	58.9	73.5	70.9
H3-pooler (Lu et al., 2023)	94.2	84.1	57.7	60.5	82.1	-
Ours	**95.4**	**87.0**	**59.7**	**62.9**	**84.9**	**82.0**

Table 2: Comparison to previous methods on the six long document classification datasets with pre-training. Bold: best scores in each column.

Methods		Hyperpartisan	20News	AMZ	BOOK	ECtHR	Essays
Full Model		95.4	87.0	59.7	62.9	84.9	82.0
Tree Structure	Removing CTT	91.3	83.6	54.8	58.8	81.8	78.9
	Removing DTT	90.9	83.1	54.1	58.7	81.7	78.2
	Removing Entire Tree Structure	88.4	78.5	47.3	52.4	77.3	73.7
Removing GAT		88.8	78.9	47.4	50.9	77.8	74.4
Removing the Bidirectional Propagation		87.9	75.9	45.2	48.9	76.1	72.7

Table 3: Ablation experiment results on the six long document classification datasets.

classifying long text. The most significant degradation occurs when the entire tree structure is replaced, especially in the multi-label classification tasks (BOOK, ECtHR, and Essays), indicating that hierarchical document modeling is important for maintaining the structural integrity and contextual coherence of the long documents.

Effectiveness of the Graph Structure. As shown in Table 3, removing the GAT module results in performance declines across all datasets, with a more pronounced impact on tasks requiring relational information, such as multi-label classification. This suggests that the GAT module is essential for capturing relationships between different elements within a document, which allows the model to leverage dependencies and interactions that are important for accurate classification. The attention mechanisms provided by the GAT module help focus on important features and connections, which are particularly important for the tasks involving complex relational data.

Effectiveness of the Bidirectional Information Propagation. The bidirectional information propagation approach further enhances our model's effectiveness by fine-tuning feature representations, as reported in Table 3. Removing this module results in significant performance reductions across all datasets, with the most marked impact observed on the BOOK and AMZ datasets. This decline indicates that the bidirectional propagation plays a critical role in polishing the feature representations

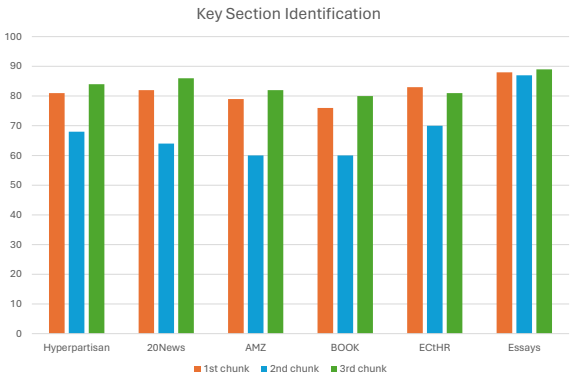

Figure 3: Visualization of the percentage of sentences selected from different chunks across all datasets.

obtained from previous layers, ensuring accurate capture of subtle and important details. By refining these features, the module helps the model better distinguish between different classes, particularly in datasets with high variability and complexity.

5.3 Key Section Identification for Long Document

In long document classification, not all sections are equally important. It is crucial to identify which parts of the documents contribute the most significant features for classification tasks. We conduct experiments by splitting each document into three chunks and calculating the number of sentences selected from each chunk. The results, shown in Figure 3, indicate that the first and third chunks consistently contain more important information

compared to the middle chunk across all datasets.

Specifically, across all datasets except for Essays, approximately 76.6% of sentences are selected from the first chunk, 83% from the third chunk, and only about 62.6% from the middle chunk. In contrast, the Essays dataset, being a questionnaire-style document, distributes important information more evenly, as each question provides different levels of information with equal importance. These observations highlight the importance of effectively capturing content at both the start and end of documents to enhance classification performance in long document tasks.

6 Related Work

Long documents present unique challenges. As the document length increases, maintaining context becomes increasingly difficult, which makes the task substantially more complex compared to short text classification (Liu et al., 2023). Early methods relied on feature extraction techniques, where document length was not a significant issue. However, this changed with deep learning approaches using CNNs and RNNs that were implemented at different semantic levels, including character, word, and sentence levels (Tang et al., 2015; Yang et al., 2016). CNNs often focus on local dependencies (Gao et al., 2018; Liu et al., 2018), while RNNs (He et al., 2019; Khandve et al., 2022) are built for long-range dependencies. Transformer-based models, such as BERT (Devlin et al., 2019) and XLNet (Yang et al., 2019), have since come to dominate, but are limited by the number of tokens they can process, which poses a challenge for handling long documents. Methods for modifying transformer architectures to handle long documents include recurrent Transformers and sparse attention Transformers (Dai et al., 2022). Transformer-XL (Dai et al., 2019), a recurrent approach, introduced a segment-level recurrence mechanism and a positional encoding scheme, which enabled the model to capture long-term dependencies more effectively. Another approach, ERNIE-Doc (Ding et al., 2021), incorporated a continuous multi-segment attention mechanism and entity-aware pre-training to capture comprehensive contextual information across longer texts. Alternatively, Sparse Transformers (Child et al., 2019) reduced the computational complexity of self-attention by selectively focusing on a subset of relevant tokens rather than all tokens in a sequence. Reformer (Kitaev et al., 2020) then improved the efficiency of Transformers by using locality-sensitive hashing for sparse attention and reversible layers to reduce memory usage. Besides the aforementioned approaches, methods such as Longformer (Beltagy et al., 2020) and Big Bird (Zaheer et al., 2020) used a combination of local and global attention mechanisms to reduce computational complexity of standard self-attention. Recently, Lu et al. (2023) used SSM to address the computational challenges (quadratic complexity in self-attention) caused by processing long sequences with traditional transformers. They introduced an SSM-pooler model, which incorporates a max pooling layer between each SSM block. This design allows the model to automatically extract important information from nearby inputs at each level and reduce the input length to half of that in the previous layer, which significantly accelerates the speed of both training and inference. They compared their model with self-attention-based models and achieved comparable performance while being, on average, 36% more efficient.

7 Conclusion

In this paper, we address challenges of long document classification by leveraging a novel graph-tree structure. By integrating syntax trees for sentence encodings and document graphs for comprehensive document encodings, our approach captures both fine-grained syntactic relationships and broader contextual dependencies. Using Tree Transformers and GAT ensures accurate modeling of hierarchical relationships within documents. Additionally, our bidirectional information propagation technique enhances the contextual representation, which enables a deeper understanding of content at all hierarchical levels. Notably, our approach not only overcomes the limitations of token constraints but also improves the performance and accuracy of long document classification, making it highly suitable for long document understanding. Potential extensions of this work could involve incorporating external knowledge through the integration of knowledge graphs. By linking document content with relevant external information, the model can further enhance its understanding and context-awareness and it would open new avenues for applications and improve the model's versatility across various domains.

Limitations

Requiring two forward passes and parsing for the tree-structured transformers increases the time required compared to the other models. This computational overhead should be taken into account when considering the deployment and scalability of the proposed models in practical applications. However, with label-wise attention cutoff values, some sentence and word nodes are pruned, which reduces the computational times significantly and the model takes similar time compared to the BERT-based model for document classification task. Still, with some parallelization in the model implementation, the computational time can be reduced.

Ethics Statement

We are using the publicly-available datasets, and we do not see any ethics issues in this paper.

References

Mahtab Ahmed, Muhammad Rifayat Samee, and Robert E. Mercer. 2019a. You only need attention to traverse trees. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 316–322, Florence, Italy. Association for Computational Linguistics.

Mahtab Ahmed, Muhammad Rifayat Samee, and Robert E Mercer. 2019b. You only need attention to traverse trees. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 316–322.

Muhammad Al-Qurishi. 2022. Recent advances in long documents classification using deep-learning. In *Proceedings of the 5th International Conference on Natural Language and Speech Processing (ICNLSP 2022)*, pages 107–112, Trento, Italy. Association for Computational Linguistics.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *Advances in NIPS 2016 Deep Learning Symposium*.

David Bamman and Noah A Smith. 2013. New alignment methods for discriminative book summarization. *arXiv preprint arXiv:1305.1319*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *Preprint*, arXiv:2004.05150.

Tom Brown, Benjamin Mann, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Ilias Chalkidis, Xiang Dai, Manos Fergadiotis, Prodromos Malakasiotis, and Desmond Elliott. 2022. An exploration of hierarchical attention transformers for efficient long document classification. *arXiv preprint*.

Ilias Chalkidis, Manos Fergadiotis, Dimitrios Tsarapatsanis, Nikolaos Aletras, Ion Androutsopoulos, and Prodromos Malakasiotis. 2021. Paragraph-level rationale extraction through regularization: A case study on European court of human rights cases. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 226–241, Online. Association for Computational Linguistics.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *Preprint*, arXiv:1904.10509.

Xiang Dai, Ilias Chalkidis, Sune Darkner, and Desmond Elliott. 2022. Revisiting transformer-based models for long document classification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7212–7230, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

SiYu Ding, Junyuan Shang, Shuohuan Wang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2021. ERNIE-Doc: A retrospective long-document modeling transformer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2914–2927, Online. Association for Computational Linguistics.

Zican Dong, Tianyi Tang, Lunyi Li, and Wayne Xin Zhao. 2023. A survey on long text modeling with transformers. *arXiv preprint arXiv:2302.14502*.

Daniel Y Fu, Tri Dao, Khaled Kamal Saab, Armin W Thomas, Atri Rudra, and Christopher Re. 2023. Hungry hungry hippos: Towards language modeling with state space models. In *The Eleventh International Conference on Learning Representations*.

Shang Gao, Arvind Ramanathan, and Georgia Tourassi. 2018. Hierarchical convolutional attention networks for text classification. In *Proceedings of the Third Workshop on Representation Learning for NLP*, pages 11–23, Melbourne, Australia. Association for Computational Linguistics.

Jun He, Liqun Wang, Liu Liu, Jiao Feng, and Hao Wu. 2019. Long document classification from local word glimpses via recurrent attention learning. *IEEE Access*, 7:40707–40718.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, page 507–517, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Snehal Ishwar Khandve, Vedangi Kishor Wagh, Apurva Dinesh Wani, Isha Mandar Joshi, and Raviraj Bhuminand Joshi. 2022. Hierarchical neural network approaches for long document classification. In *Proceedings of the 2022 14th International Conference on Machine Learning and Computing*, ICMLC '22, page 115–119, New York, NY, USA. Association for Computing Machinery.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. 2019. SemEval-2019 task 4: Hyperpartisan news detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 829–839, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *International Conference on Learning Representations*.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In Armand Prieditis and Stuart Russell, editors, *Machine Learning Proceedings 1995*, pages 331–339. Morgan Kaufmann, San Francisco (CA).

Liu Liu, Kaile Liu, Zhenghai Cong, Jiali Zhao, Yefei Ji, and Jun He. 2018. Long length document classification by local convolutional feature aggregation. *Algorithms*, 11(8).

Tengfei Liu, Yongli Hu, Boyue Wang, Yanfeng Sun, Junbin Gao, and Baocai Yin. 2023. Hierarchical graph convolutional networks for structured long document classification. *IEEE Transactions on Neural Networks and Learning Systems*, 34(10):8071–8085.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Peng Lu, Suyuchen Wang, Mehdi Rezagholizadeh, Bang Liu, and Ivan Kobyzev. 2023. Efficient classification of long documents via state-space models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6559–6565, Singapore. Association for Computational Linguistics.

Agnes Lydia and Sagayaraj Francis. 2019. Adagrad—an optimizer for stochastic gradient descent. *Int. J. Inf. Comput. Sci*, 6(5):566–568.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Raghavendra Pappagari, Piotr Zelasko, Jesús Villalba, Yishay Carmiel, and Najim Dehak. 2019. Hierarchical transformers for long document classification. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 838–844.

Hyunji Park, Yogarshi Vyas, and Kashif Shah. 2022. Efficient classification of long documents using transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 702–709, Dublin, Ireland. Association for Computational Linguistics.

James W Pennebaker and Laura A King. 1999. Linguistic styles: language use as an individual difference. *Journal of personality and social psychology*, 77(6):1296.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432, Lisbon, Portugal. Association for Computational Linguistics.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2020. Long range arena: A benchmark for efficient transformers. In *International Conference on Learning Representations*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. LLaMa 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: generalized autoregressive pretraining for language understanding. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA. Curran Associates Inc.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big Bird: transformers for longer sequences. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.