Sequence Graph Network for Online Debate Analysis

Quan Mai, [1] Susan Gauch, [1] Douglas Adams, [2] Miaoqing Huang [1]
[1]*Department of Electrical Engineering and Computer Science,* [2]*Department of Sociology and Criminology*
University of Arkansas
Fayetteville, Arkansas, USA
{quanmai, sgauch, djadams, mqhuang}@uark.edu

***Abstract*—Online debates involve a dynamic exchange of ideas over time, where participants need to actively consider their opponents' arguments, respond with counterarguments, reinforce their own points, and introduce more compelling arguments as the discussion unfolds. Modeling such a complex process is not a simple task, as it necessitates the incorporation of both sequential characteristics and the capability to capture interactions effectively. To address this challenge, we employ a sequence-graph approach. Building the conversation as a graph allows us to effectively model interactions between participants through directed edges. Simultaneously, the propagation of information along these edges in a sequential manner enables us to capture a more comprehensive representation of context. We also introduce a Sequence Graph Attention layer to illustrate the proposed information update scheme. The experimental results show that sequence graph networks achieve superior results to existing methods in online debates.**

***Keywords*-Graph neural networks; dialog modeling; sequence graph network; online debates.**

I. Introduction

Online debate has become an integral part of our digital age, transforming the way we engage in discourse and exchange ideas. In social media platforms (e.g., Facebook, Twitter (currently X), etc.), individuals from diverse backgrounds and geographical locations converge to discuss and deliberate on a wide array of topics, ranging from politics and ethics to music and science. Debating with a wide range of debaters requires participants to research and present well-informed arguments, encourages critical thinking, and challenges preconceived notions.

Like other forms of debate, online discussions are contingent on the flow of time (temporal dependency); each subsequent comment relies on the content of the previous comment it responds to. Participants interactively promote their point while countering the opponent's [4]. Within a turn, debaters employ a variety of strategies, each of which plays a crucial role in determining the outcome of the debate. These strategies involve either directly addressing the opponent's argument, presenting their own viewpoint, or skillfully combining both tactics. The latter approach often appears to be the most effective, allowing the debater to simultaneously achieve both objectives during their turn. However, one cannot always adopt that strategy as it depends on their position in the debate. For instance, if a debater is the first speaker in a debate, their primary task is to present their own ideas coherently and logically, as they do not have the opportunity to directly counter their opponent's arguments at this stage. In such a scenario, the debater's effectiveness lies in the clarity and persuasiveness of their presentation, making it challenging for the opposing side to refute their position. These strategies are also discussed in [4], which examined the dynamics of information flow within online debates.

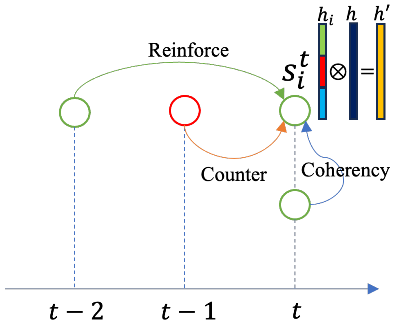


Figure 1. A "what-should-we-mention" information flow scheme that mimics the interaction process of a debater. At each time step t, the node features are updated by considering their peer nodes from the same turn and the connected nodes from previous turns, using Directed Graph Attention Network layers. Nodes associated with different debaters are colored differently. Each type of edge (colored arrows) contributes a corresponding representation, collectively forming $\mathbf{h}_i$. The node's utterance embedding $\mathbf{h}$ and the interaction representation $\mathbf{h}_i$ are used to update the node feature $\mathbf{h}'$.

As the argument process is temporally dependent, Recurrent Neural Networks (RNNs), such as Long Short Term Memory (LSTM) [9] and Gated Recurrent Unit (GRU) [13], have been one of the most widely used techniques in argument-winning research as well as dialog extraction. Several studies employ RNNs as the encoder for utterances [5] [7] [10], leveraging their capacity to capture sequential dependencies and relationships within textual data. In addition to encoding individual utterances, sequence networks are employed to encode entire conversations by sequentially processing the arguments [11].

In a debate, however, participants engage in interactive turn-by-turn rebuttals to counter their opponents' arguments, and sequencing the entire conversation fails to capture this dynamic interaction. In order to model the process of dialogical argumentation, [10] use a co-attention network to capture the interaction between the participants and achieve a promising performance on the prediction task. The focus of [7] is placed on identifying connections between the sentences of debaters. This approach is instrumental in capturing critical argumentative components, making it a pivotal factor for predicting the winner. The aforementioned studies compute "attention scores" for each pair of sentences belonging to two participants in order to assess the *relevance* of one sentence to another.

An alternative method for capturing these interaction dynamics is through the use of graphs. Graphs are an effective way to represent relationships and dependencies among entities, making them suitable for a wide range of applications, including social networks and recommendation systems [17]–[19]. The connection between two components of an argument can be effectively represented by a link (or edge) within the graph. Graphs can also serve as input to Graph Neural Networks (GNNs) for capturing the contextual information within the conversation. In their work, [12] employ a heterogeneous graph to represent the relationships among entities discussed in multi-party dialogues. In order to model the relationships between argument pairs, [5] incorporate intra-passage and cross-passage links to interconnect sentence nodes. Subsequently, they employ a Graph Convolutional Network (GCN) [15] for efficient information propagation.

Traditional GNNs, including GCNs and Graph Attention Networks (GAT) [3]), may not effectively capture the temporal dynamics within a conversation, particularly in a debate scenario in which participants engage in interactive exchanges to counter arguments or defend their own viewpoints. To tackle this challenge, we integrate the strengths of both RNNs and GNNs within a unified framework. In this framework, we conceptualize the debate as a graph, where argument components are depicted as nodes, and their features undergo sequential updates, according to the turn to which they correspond. We introduce the Sequence Graph Attention (SGA) cell, which resembles the traditional RNN-cell, to capture long-range dependencies in the debate (which is treated as a sequence of subgraphs). The experimental results demonstrate that our approach can capture the interaction between debaters and outperforms state-of-the-art models in accurately predicting the winner in several online debate datasets. The code and models are available at [39].

The structure of the remainder of this paper is organized as follows: Section II describes the process of constructing a graph from a debate. In Section III, we introduce our proposed framework. The effectiveness of this method is evaluated in Section IV. Section V reviews some relevant literature. Finally, Section VI provides a summary of our findings and discusses potential avenues for future work.

II. Preliminary

Before describing the details of the proposed method, we first give a brief introduction to how we construct a graph for an online debate.

A. Debate Format

Our primary focus lies in online debates wherein the victor emerges through the collective votes of an audience or a panel of judges. These debates adhere to the Oxford-style format, featuring two participants representing opposing viewpoints —one in favor of the claim (Pros) and the other in opposition (Cons) — who alternate in presenting their arguments on a given topic. After the debate, a winner is declared, unless a tie occurs. In this study, we define a *turn* as each instance when a debater presents their argument, and a *round* represents the stage in which opposing sides provide their arguments. Consequently, round 0 consists of turn 0 and turn 1, round 1 consists of turn 2 and turn 3, and so forth.

B. Debate-to-Graph construction

Given a debate that contains a total of N sentences, a directed, unweighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{H})$ is constructed based on sentences and their relationships (Figure 2). Sentences in the debate are represented by a set of nodes $\mathcal{V}$ ($|\mathcal{V}| = N$), and a node attribute matrix $\mathcal{H} \in R^{N \times D}$, defined by D-dimensional embedding vectors for each of the sentences. Sentences in the debate may be interconnected and these interconnections are represented by $\mathcal{E}$, the set of edges in the graph.

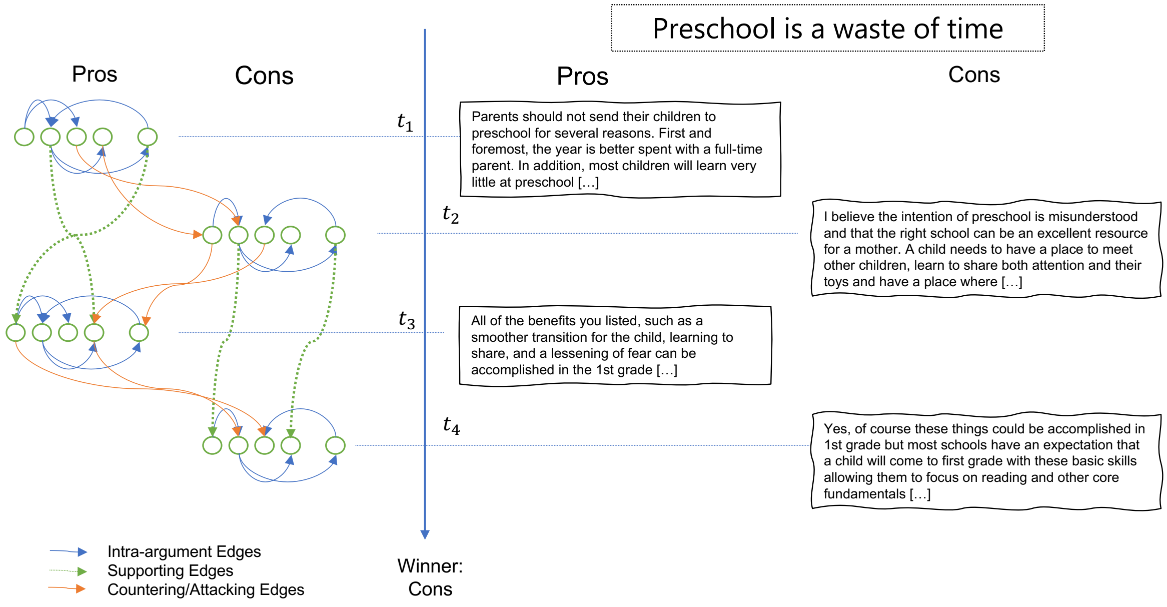
Edge types: We define three different types of edges to elucidate the participants' strategies throughout the debate. Each type is categorized based on the turn it corresponds to and the strategic role it plays. In Section III, we will delve into how each type contributes to node feature aggregation.

1) Logical and Coherent Edges: These edges emphasize the participants' ability to construct logical and coherent arguments within their turn.
2) Reinforcement Edges: These edges serve to strengthen the points previously made by the debater in their previous rounds. We will interchangeably use the terms *reinforcement edges* and *supporting edges*.
3) Counterargument Edges: These edges highlight the participants' skill in countering their opponents' arguments effectively.

Intra-argument Links These edges connect sentences of the same turn. During a turn, edges are constructed based on the relative position among sentences. These *Logical and Coherent* edges capture coherency in an argument turn. Given two sentences, denoted as s_i^t and s_j^t, both belonging to turn t, we establish an edge e_{ij}^{inter} from s_j^t to s_i^t if the positional difference $\mathcal{D}$ between them is within a specified distance threshold d.

$$e_{ij}^{inter} = \begin{cases} 1 & \text{if } \mathcal{D}(s_i^t, s_j^t) \leq d \\ 0 & \text{otherwise} \end{cases}$$

Cross-argument Links These edges interconnect sentences that belong to different turns and are categorized into two types: *Reinforcement* and *Counterargument* edges. The former connects nodes belonging to the same debater whereas the latter connects nodes belonging to different debaters. For example, nodes in the 3rd turn are connected to nodes from the 1st turn through *Reinforcement* edges and are also linked with their opponent's nodes from the 2nd turn. Unlike intra-argument edges that rely on the relative positions of sentences, cross-argument edges are established using semantic textual similarity between sentences. In this work, we use cosine similarity S_c to capture the semantic relationship of texts.



Parents should not send their children to preschool for several reasons. First and foremost, the year is better spent with a full-time parent. In addition, most children will learn very little at preschool [...]

I believe the intention of preschool is misunderstood and that the right school can be an excellent resource for a mother. A child needs to have a place to meet other children, learn to share both attention and their toys and have a place where [...]

All of the benefits you listed, such as a smoother transition for the child, learning to share, and a lessening of fear can be accomplished in the 1st grade [...]

Yes, of course these things could be accomplished in 1st grade but most schools have an expectation that a child will come to first grade with these basic skills allowing them to focus on reading and other core fundamentals [...]

Figure 2. Graph Construction from Debate: Nodes establish connections through three distinct edge types, indicated by colored arrows. Intra-argument edges (blue) link nodes within the same turn, reinforcement edges (green) connect nodes from the same debater across different turns, while countering edges (orange) connect nodes from a debater to their opponent's, illustrating counter-argumentation. The sample debate is taken from data collected by [1].

An edge e_{ij} links 2 nodes v_i and v_j if their similarity score $S_c(\mathbf{h}_i, \mathbf{h}_j)$ meets a threshold value S_{th}

$$e_{v_i,v_j} = \begin{cases} 1 & \text{if } S_c(\mathbf{h}_i, \mathbf{h}_j) \geq S_{th} \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbf{h}_i$ and $\mathbf{h}_j$ are i^{th} and j^{th} rows in $\mathcal{H}$, representing embedding vectors of sentences v_i and v_j, respectively. S_{th} serves as a crucial hyper-parameter for evaluating the influence of participant interactions on the debate's outcome. An alternative approach is to employ the top k similarities, allowing each node to establish connections with up to k cross-argument nodes that possess the highest similarity scores. We will evaluate the effectiveness of each approach on the predictive performance in Section IV. It is important to note that cross-argument edges consistently flow from nodes in previous turns to nodes in subsequent turns; there is no reverse direction.

III. Proposed Method

A. Utterance Encoder

We encode each sentence using pre-trained sentence embedding (Sentence Transformer (SBERT)) [2]. In preliminary work, we found that this approach works better than using GloVe [6] word embeddings and a bidirectional LSTM to encode semantic vectors for sentences. This step gives us the sentence embedding matrix $\mathcal{H}$, in which each row $\mathbf{h}_i$ is an embedding vector for sentence s_i.

Turn Embeddings: Participants employ distinct strategies during different debate turns. For instance, in the initial round consisting of two turns, the first participant presents their perspective on the topic while the second participant challenges their opponent's arguments and introduces their own viewpoint. We incorporate the temporal turn information into the node features by concatenating it with the sentence embedding $\mathbf{h}_i$. We opt for a 30-dimensional embedding vector $\mathbf{h}_{it} \in R^{30}$ to represent the turn information for each node.

$$\mathbf{h}_i = \mathbf{h}_i \| \mathbf{h}_{it} \tag{1}$$

Let B denote the number of dimensions of the embedding vector of a sentence from SBERT, then $D = B + 30$.

B. Information flow

Graph Attention Layer: We employ a Graph Attention Network (GAT) [3] layer to update the node representation. The attention mechanism allows GAT to focus on and weigh the importance of different neighbors when aggregating information for each node, called the "attention score". We are motivated to use GAT in our model because, intuitively, not all sentences in the debate carry equal importance. One can detect the opponent's argumentative "vulnerable region" [7] and effectively counter it to win the debate. This layer takes as input a set of A ($A \leq N$) node features $\mathbf{h} \in \mathbb{R}^{A \times D}$ and produces a new set of node features $\mathbf{h}' \in \mathbb{R}^{A \times D'}$ ($D' < D$). The attention score of sentence j to sentence i is computed as:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}\mathbf{h}_i \| \mathbf{W}\mathbf{h}_j]))}{\sum_{k \in \mathcal{N}} \exp(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}\mathbf{h}_i \| \mathbf{W}\mathbf{h}_k]))}$$

where $\mathbf{W} \in \mathbb{R}^{D \times D'}$ and $\mathbf{a} \in \mathbb{R}^{2D'}$ are trainable weight matrix and vector of the layer. The output features of node i is the weighted sum of the features of its neighboring node set $\mathcal{N}_i$:

$$\mathbf{h}'_i = \sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W}\mathbf{h_j}$$

In this work, we employ three distinct GAT layers, each responsible for aggregating information from a specific type of edge. We refer to these layers as GATI (intra-argument edge), GATC (counterargument edge), and GATS (supporting edge). At each turn, the GAT layer processes a specific set of input node features and produces a new set of features, called *interaction* representation of each sentence:

$$\mathbf{h}_I^t = \text{GATI}(\mathbf{h}_{\mathcal{I}_t}; \mathbf{a}^I, \mathbf{W}^I) \tag{2}$$

$$\mathbf{h}_C^t = \text{GATC}(\mathbf{h}_{\mathcal{J}_t}; \mathbf{a}^C, \mathbf{W}^C) \tag{3}$$

$$\mathbf{h}_S^t = \text{GATS}(\mathbf{h}_{\mathcal{K}_t}; \mathbf{a}^S, \mathbf{W}^S) \tag{4}$$

where $\mathbf{a}^*$ and $\mathbf{W}^*$ are vectors and matrices associated with each layer. Here, we have three sets of node features: $\mathbf{h}_{\mathcal{I}_t}$, $\mathbf{h}_{\mathcal{J}_t}$, and $\mathbf{h}_{\mathcal{K}_t}$, each corresponding to distinct node sets:

- $\mathcal{I}_t$ represents the set of nodes that pertain to the same time step, encompassing nodes within the current turn. $\mathbf{h}_{\mathcal{I}_t} = \{\mathbf{h}_1^t, \mathbf{h}_2^t, \mathbf{h}_3^t, ...\}$ denotes features matrix of a set of nodes at time t.
- $\mathcal{K}_t$ comprises nodes from time steps $t-2$ and t, all originating from the same debater and exhibiting a supportive relationship. This set characterizes argumentative enhancement or promotion. Note that the set of node features at time $t-2$ are **updated** in turn $t-2$. Therefore, $\mathbf{h}_{\mathcal{J}_t} = \{\mathbf{h}_1^{\prime t-2}, \mathbf{h}_2^{\prime t-2}, ..., \mathbf{h}_1^t, \mathbf{h}_2^t, ...\}$ denotes the updated features matrix of a set of nodes at times $t-1$ and utterance matrix of nodes at t.
- In contrast, $\mathcal{J}_t$ encompasses nodes from time steps $t-1$ and t and signifies an adversarial relation, capturing how a debater challenges an opponent's position by considering nodes from the opponent's previous turn $(t-1)$. Because nodes feature at time $t-1$ are updated, $\mathbf{h}_{\mathcal{K}_t} = \{\mathbf{h}_1^{\prime t-2}, \mathbf{h}_1^{\prime t-2}, ..., \mathbf{h}_1^t, \mathbf{h}_2^t, ...\}$.

a) Sequential Update: The node features are updated sequentially using a temporal attention mechanism. Information propagation occurs along *directed* edges, and the features of nodes at time t are updated based on their neighboring nodes from the same turn (via intra-argument edges) as well as nodes from previous turns (via cross-argument edges) (Figure 1). This information flow scheme illustrates the cognitive process of a debater during their turn, as they must consider the opponent's previous arguments, formulate counterarguments, reinforce their own points, and even introduce new ideas. The node features updated at time t serve as the input when updating node features at times $t+\tau$ ($\tau \in \{1, 2\}$). This process shares similarities with traditional RNNs like LSTM and GRU. However, it is important to note that our work focuses on handling a specific subset of nodes at each timestep. This distinction sets us apart from Gated Graph Sequence Neural Networks [8] that process the entire graph as input at each timestep. Similar to an RNN-Cell, that operates on a single input element at each time step and generates output that serves as a hidden feature for subsequent times, we introduce the SGA layer to manage the processing of a specific subset of nodes at time t. The entire debate graph is processed sequentially subgraph-by-subgraph.

Given a debate $\mathcal{S}$ that has T turns: $\mathbb{S} = \{S_t; t \in [0, T-1]\}$, $S_t = \{s_j^t; j \in [0, M_t - 1]\}$ denotes a debate turn consisting of M_t sentences s_j^t. It is noticeable that $N = \sum_{t=0}^{T-1} M_t$. Let $\mathbf{h}_j^t$ the utterance embedding of the sentence s_j (from 1), the new node feature h_j' is calculated using the SGA layer which executes the following operations (we discard the superscript t for readability):

$$h_j' = SGA(\mathbf{h}_j, \mathbf{h}_{\mathcal{I}}, \mathbf{h}_{\mathcal{J}}, \mathbf{h}_{\mathcal{K}}) = \mathbf{h}_j \otimes \mathbf{h}_j^{\text{X}} \tag{5}$$

where $\otimes$ is the update operator using GRU operations [13]. The $\mathbf{h}_j^{\text{X}}$ denotes the interaction representation feature at time t, encompassing intra-argument coherency, counterarguments against the opponent's points, and reinfordcement of the debater's previous statements. It is calculated by concatenating the node features produced by three component GAT layers (equations 2, 3, 4):

$$\mathbf{h}_j^{\text{X}} = \mathbf{h}_j^{\text{GATI}} \mid\mid \mathbf{h}_j^{\text{GATC}} \mid\mid \mathbf{h}_j^{\text{GATS}} \tag{6}$$

It is important to observe that during the initial turn, denoted as $t = 0$, there are no counterarguments in the debater's thoughts. As a result, we initialize $\mathbf{h}_j^{\text{GATC}\ 0}$ to be equal to 0. Additionally, a debater does not introduce a reinforcing argument until their second round (or when $t \geq 2$). Consequently, both $\mathbf{h}_j^{\text{GATS}\ 0}$ and $\mathbf{h}_j^{\text{GATS}\ 1}$ are set to 0 during this period. The updated node features h_j' are then employed to update the attributes of nodes in subsequent turns.

C. Readout Layer

Once all the node features have been updated, we employ a readout layer to "summarize" the ideas presented by each participant during the debate. For each debater, we select a set of top r (e.g., $r = 3$) *representatives*, which are used as input for the prediction classifier. The process of selecting these representative nodes is determined by the highest "attention scores" generated by each GATI, GATC, and GATS layers, denoted as α_I, α_C, and α_S, respectively. During the feature update step, each node receives an attention score from its neighboring nodes. These scores emphasize the significance of a node in relation to others. The more significant a node is, the greater its contribution to a debater's overall idea. The total attention received by each node is obtained by summing up its individual attention scores. Consider a node s_l, its attention scores are:

$$\alpha_{s_l}^I = \sum_{i \in \mathcal{I}} \alpha_i, \ \alpha_{s_l}^C = \sum_{j \in \mathcal{J}} \alpha_j, \ \alpha_{s_l}^S = \sum_{k \in \mathcal{K}} \alpha_k \tag{7}$$

We opt to select the top r nodes with the highest scores for each type of attention. We then concatenate the feature vectors corresponding to these selected nodes to create a $3 \times r \times D'$-dimensional vector, where D' is the dimension of the node feature produced by SGA. The readout layer subsequently generates two "summary" vectors, each serving as a deep representation of each debater's performance during the debate.

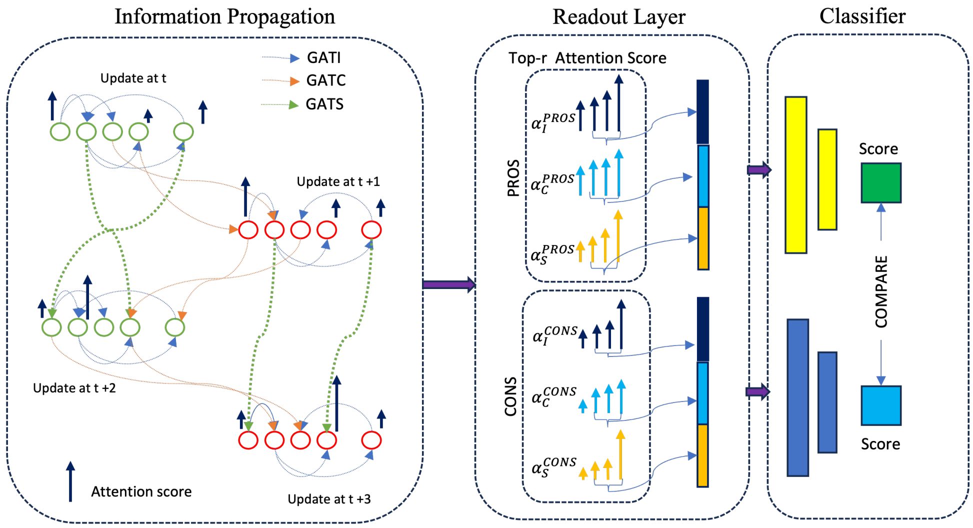


Figure 3. The proposed architecture consists of three key modules: (1) Information propagation is driven by the SGA layers, updating node features sequentially using a graph attention mechanism. (2) The readout layer identifies representative vectors associated with each debater, which are subsequently supplied as input to (3) an MLP classifier for predicting the debate winner.

D. Classification

The two vectors, $\mathbf{Q}^{PROS}$ and $\mathbf{Q}^{CONS}$ achieved by the readout layer are fed to the classifier to perform the prediction task. Each vector is mapped to a score value $c \in \mathbb{R}^1$ by linear transformation using a Fully Connected (FC) layer followed by an activation function (e.g., ReLU), Layer Norm (LN) [14] and dropout layer [24]. Let us denote a series of FC + ReLU + LN + Dropout an MLP, then

$$c^{PROS} = \mathit{MLP1}(\mathbf{Q}^{PROS})$$
$$c^{CONS} = \mathit{MLP2}(\mathbf{Q}^{CONS})$$

If the Pros side wins, we expect that $c^{PROS} > c^{CONS}$, and conversely when the Cons side wins. Here, we denote C^+ and C^- as the scores of the winner and loser, respectively. Our objective is to maximize the difference between C^+ and C^- as much as possible. To achieve this, we employ Pairwise Cross-Entropy (PCE) loss, that minimizes:

$$\mathcal{L} = \mathrm{PCE}(C^+, C^-) = \log(1 + exp(C^- - C^+)) \quad (8)$$

The network architecture is illustrated in Figure 3.

IV. Evaluation

A. Dataset

Our study is conducted on the *debate.org* dataset collected by [1]. The dataset contains 78,376 debates on controversial topics, including *abortion*, *death penalty*, *gay marriage*, and *affirmative action*. Each debate consists of multiple rounds in which two participants from two opposing sides take turns expressing their opinions. Further details can be found in [1].

a) Winning criterion: The winner is determined by the criterion of "Made more convincing arguments". We exclude debates with fewer than 5 voters and tie debates. Additionally, debates in which the winner has just one more vote than the loser are also classified as ties.

TABLE I
The number of sentences, number of Counterargument Edges, and number of Supporting Edges made by winner and loser in an argument turn. Cross-argument edges are constructed using a similarity threshold of 0.85.

	#Sentences	#Countering	#Supporting
Winner	38.6	6.96	5.93
Loser	36.1	6.78	6.64

b) Preprocessing: To study the interaction among debates, we only keep debates that have at least 3 rounds (equivalent to 6 turns). Short arguments are also eliminated, i.e., we remove debates that have fewer than 5 sentences in each round (each graph thereby has at least 30 vertices). The first 3 rounds of longer debates are used for analysis. The dataset exhibits an imbalance, with the Cons side accounting for 65% of the winners whereas the Pros side wins only 35%. To create a balanced dataset, we also use the final 3 rounds of the debates where the Pros side wins and the debate comprises more than three rounds. This data augmentation step also increases the size of the dataset.

c) Statistics: After the experimental dataset selection step, there are a total of 2,445 debates available for model training and testing. Among these debates, the Pros side wins in 1,130 debates, while the Cons side secures victory in 1,325 debates. Additional statistical information is shown in table I. Observing the table, it becomes evident that the winning side tends to produce more sentences and more counterarguments compared to the losing side. Conversely, the losing side appears to prioritize reinforcing their own ideas rather than generating a higher number of counterarguments.

B. Experimental setup

a) Data Preprocessing: We randomly split the dataset with 60% for training, 20% for validation and 20% for testing. For text normalization, we employ the following steps: (1) replacing URLs with “website”, (2) replacing all the numbers with “number”, and (3) lowercasing text. Next, we employed spaCy [23] for sentence tokenization. Sentences are then encoded by SBERT’s “all-MiniLM-L6-v2” model that transforms a sentence into a 384-dimensional vector.

b) Parameter setting: We use a similarity threshold of 0.85 for cross-argument edge construction, other approaches regarding edge construction will be further discussed in the ablation study. The intra-argument distance threshold is $d = 3$. Each node within a turn links to nodes that share a relative positive correlation within a 3-node proximity. Node features updated by each GAT layer have $D' = 32$ dimension. For the readout layer, we choose $r = 3$. We use a stack of three MLPs to transform the readout layer’s output into a score for each debater. The first layer reduces the vector from $3 \times r \times D$ to half its size. The second layer further reduces the output of the first layer by half, and the final layer maps the second output vector to a real value. We apply the tanh function to ensure the value falls within the range [-1; 1]. For hyper-parameters, we apply the dropout rate of 0.2 for all GAT layers and the classifier. Optimization is performed using Adam [16]. The batch size is 32. We run the model for 50 epochs with early stopping. The learning rate is 0.0001.

c) Other settings: Deep learning frameworks are Pytorch [21] and Pytorch Lightning [22]. We use DGL package [20] as the graph deep learning framework. The networks are trained and tested on an NVIDIA Quadro RTX 8000 GPU with 50GB of memory.

C. Comparison baselines

Given that the Cons side accounts for 52.5% of wins in the test set, it serves as the **majority baseline**, representing the best prediction one can make regardless of the input features. We compare our model’s performance to SOTAs in debate winning prediction which adopt sequence approach in their work.

a) Sequence approach: In the study by [11], they aggregate the entire discussion into a single sequence and model it using LSTM with an attention mechanism applied to the sentences, referred to as the **all-LSTM** approach. They also incorporate implicit discourse relations using the Penn Discourse Tree Bank [25] discourse structure. While their research primarily centers on the Reddit dataset [35], we apply the same methodology to our debate dataset. Additionally, we find relevance in the work of [26], denoted as **ASODP**, which shares our focus on Oxford-style debates and employs a sequential approach for debate analysis. Furthermore, [27] introduces the **DTDMN** method, designed to process pairs of conversations and predict their persuasiveness. Similarly, we present the Pros and Cons sides as inputs to facilitate comparative analysis.

TABLE II
DEBATE-WINNING PREDICTION RESULTS. THE BEST RESULTS ARE IN BOLD. (**: USING THE TOP 3 HIGHEST SIMILARITY SCORES TO CONSTRUCT CROSS-ARGUMENT EDGES, *: USING A THRESHOLD VALUE OF 0.85 TO CONSTRUCT CROSS-ARGUMENT EDGES).

Models	Acc.	F1
Majority Baseline	0.525	
Sequence Baseline		
all-LSTM	0.635	0.563
ASODP	0.656	0.623
DTDMN	0.660	0.625
Graph Baseline		
GAT	0.541	0.472
GGNN	0.565	0.522
Sequence Graph Baseline		
Graphflow	0.645	0.620
SGA		
w/o GATI	0.621	0.523
w/o GATC	0.562	0.495
w/o GATS	0.629	0.534
FULL MODEL		
*S = 0.85	0.654	**0.667**
**k = 3	**0.675**	0.625

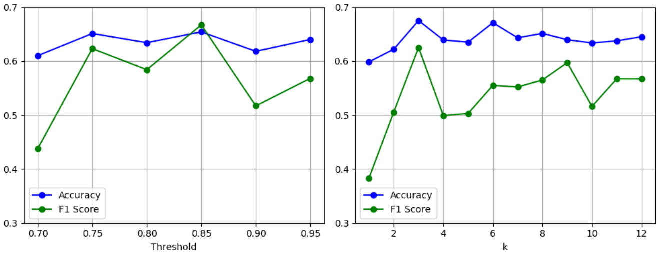


Figure 4. Impact of cross-argument construction values on network performance. Left: Edge construction using a threshold value. Right: Edge construction using top-k highest values.

b) Graph approach: To highlight the significance of processing the debate on a turn-by-turn basis, we introduce two baseline models for graph analysis. The first baseline employs a 2-layer **GAT** network, while the second baseline utilizes a **GGNN**. These GNNs serve as information aggregators and feature extractors for the debate graph, simultaneously processing all nodes in the graph (and repeating this process 6 times, corresponding to 6 turns in the case of GGNN). In the case of GAT, the initial layer transforms the input into 64-dimensional vectors, and the subsequent layer maps the output from the first layer to 32-dimensional features. In GGNN, we also utilize a 32-dimensional output feature size to align with the output feature size of our SGA layer. To summarize the node features for each debater, we introduce a mean readout operation.

c) Temporal graph approach: Since no other sequential graph approach exists for debate winning prediction, we adopt the information flow method proposed in [33] (**Graphflow**), initially designed for machine comprehension. We utilize the output of the RGNN layer from the final turn, feeding it into the MLP layer for the prediction task.

D. Experimental results

The evaluation results are presented in Table II. The sequence baselines (all-LSTM, ASODP, DTDMN) all perform similarly, with DTDMN producing the best accuracy of this group at 66.0%. The Graph baselines perform more poorly, with the highest accuracy, 56.5% produced by SSGN. Graphflow, boasting an accuracy of 64.5%, outperforms traditional graph approaches. However, it still trails behind the robust benchmarks set by sequential approaches such as ASODP and DTDMN. Our full model, SGA with k=3, outperforms all baselines with an accuracy of 67.5%, a 1.5% absolute (2.3% relative) improvement over DTDMN. The F1-score, achieved by constructing cross-argument edges with a threshold of 0.85, significantly outperforms the baselines. It reaches 66.7%, representing a 4.2% absolute (or 6.7% relative) improvement over DTDMN. We thus demonstrate that we outperform state of the art models for this dataset.

The performances of all-LSTM and DTDMN are diminished when applied to the *debate.org* dataset. This can be attributed to a fundamental distinction between the two domains. In the context of *debate.org*, the ultimate determination of the winner is not based on subjective criteria but rather relies on the judgments of a panel of judges or the voters. The voters place substantial emphasis on the debaters' ability to rigorously address and counter their opponents' reasoning. Furthermore, they favor debaters who engage in high-quality and dynamic interactions throughout the debates.

a) Sequence matters: The results show a significant superiority of sequence-based baselines over graph-based ones when applied to the debate dataset. This highlights the critical significance of adopting a sequential approach, where the debate is processed turn-by-turn, rather than relying solely on graph-based methodologies.

b) Counter-argument is crucial: We extended our analysis by performing an ablation study to assess the individual impact of each GAT layer on our proposed SGA model. We observe that when we omit the counter-argument edges, the reduction in network performance was more significant compared to scenarios where we exclude either GATI or GATS layers. Specifically, accuracy drops by 11.3%, in contrast to 5.4% and 4.6%, respectively. This outcome can be elucidated by considering that if a debater disregards the opponent's remarks from the preceding turn, their persuasive ability may diminish in the eyes of the voters or judges. In essence, acknowledging and responding to counter-arguments plays a pivotal role in constructing compelling arguments in a debate context.

E. Impact of graph parameters

We conduct a detailed analysis of the impact of graph construction parameters, such as S_{th} and k, on the network's performance (Figure 4). In the context of employing a similarity threshold, it is noteworthy that a threshold value of 0.85 yields the highest performance in terms of accuracy and F1-score.

Regarding the top-k approach, it is worth highlighting that while $k = 3$ achieves the highest accuracy, as well as highest F1-score. These insights into parameter effects contribute to a deeper understanding of how to optimize network performance for specific objectives and trade-offs.

V. Related work

Graph Neural Networks (GNNs) have proven to be powerful tools for harnessing insights into, and making predictions on, data structured as graphs, particularly in the realm of Natural Language Processing (NLP). Within NLP, GNNs have been applied to a wide spectrum of tasks including, but not limited to, dependency parsing [29], sentiment analysis [30] [31], and semantic understanding [15]. In recent developments, researchers in NLP have extended GNNs by integrating them with RNNs to enable sequential processing of graph-structured data. Notably, [32] introduced a graph-to-sequence methodology for the AMR-to-text generation task, wherein they construct an Abstract Meaning Representation (AMR) graph and progressively update the entire graph during sequential generation. Furthermore, [33] made significant strides in the domain of machine comprehension by incorporating conversation history into their model. They adopt a graph-based approach, constructing a graph that evolves with each conversational turn. While our work shares a commonality in the sequential update of subgraphs, it is important to emphasize that the implementation details diverge significantly. Researchers have explored temporal graph approaches for tasks like traffic flow forecasting [36] [37] and skeleton-based action recognition [38]. However, the utilization of sequence graph approaches in conversation analysis, particularly within online debate and argumentative analysis contexts, remains relatively unexplored.

VI. Conclusion and future work

In conclusion, the task of modeling online debates, characterized by the dynamic exchange of ideas, is a challenging endeavor. To tackle this complexity, we introduced a novel approach using sequence-graph modeling. By representing conversations as graphs, we effectively captured the interactions among participants through directed edges, while the sequential propagation of information along these edges enriched our understanding of context. Our incorporation of the SGA layer demonstrated the efficacy of our information update scheme. Our experimental results demonstrate the success of sequence graph networks in outperforming existing methods when applied to Oxford-style online debate dataset.

The proposed method not only advances the ability to model dynamic discussions but also highlights the potential of sequence-graph approaches for a wide range of tasks involving sequential interactions and context-rich data. As online debates continue to evolve, the techniques presented in this paper offer valuable insights into improving our understanding of complex conversational dynamics.

While the proposed method has demonstrated promising results in predicting debate outcomes, it does exhibit certain limitations. Firstly, the construction of cross-argument edges relies solely on similarity scores. While this approach may suffice for reinforcing connections, it may not consistently identify valid counterarguments. High similarity scores between two sentences do not guarantee a counterrelation. Secondly, the method overlooks the utilization of argument structures. The

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

intra-argument links primarily capture temporal relationships by connecting adjacent sentences. However, this approach fails to account for potential relationships between sentences that are distant within an argument turn. There is room for improvement by incorporating pre-trained models that account for argumentative structures. For instance, [26] enhanced predictability on debate datasets by integrating argument structure introduced by [34].

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Award # OIA-1946391, "Data Analytics that are Robust and Trusted (DART)".

REFERENCES

[1] E. Durmus and C. Cardie, "A corpus for modeling user and language effects in argumentation on online debating," Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Italy, pp. 602–607, 2019.

[2] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, pp. 3982–3992, 2019.

[3] P. Veličković et al., "Graph attention networks," Proceedings of the 6th International Conference on Learning Representations, Canada, 2018.

[4] J. Zhang, R. Kumar, S. Ravi, and C. Danescu-Niculescu-Mizil, "Conversational flow in Oxford-style debates," Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, California, pp. 136–141, 2016.

[5] J. Bao et al., "Argument pair extraction with mutual guidance and inter-sentence relation graph," Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 3923–3934, 2021.

[6] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Qatar, pp. 1532–1543, 2014.

[7] Y. Jo et al., "Attentive interaction model: Modeling changes in view in argumentation," in Proc. of the 2018 Conf. of the North American Chapter of the Assoc. for Computational Linguistics: Human Language Technologies, Vol. 1, pp. 103-116, 2018.

[8] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," Proceedings of the 4th International Conference on Learning Representations, Puerto Rico, 2016.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[10] L. Ji et al., "Incorporating Argument-Level Interactions for Persuasion Comments Evaluation using Co-attention Model," Proceedings of the 27th International Conference on Computational Linguistics, USA, pp. 3703–3714, 2018.

[11] C. Hidey and K. McKeown, "Persuasive influence detection: The role of argument sequencing," Proceedings of the AAAI Conference on Artificial Intelligence, pp. 5173–5180, 2018.

[12] H. Chen, P. Hong, W. Han, N. Majumder, and S. Poria, "Dialogue relation extraction with document-level heterogeneous graph attention networks," Cognitive Computation, vol. 15, no. 2, pp. 793-802, 2023.

[13] K. Cho et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Qatar, pp. 1724–1734, 2014.

[14] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," arXiv preprint arXiv:1607.06450, 2016.

[15] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," Proceedings of the 5th International Conference on Learning Representations, France, 2017.

[16] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in 3rd International Conference on Learning Representations, USA, 2015.

[17] S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, "Graph neural networks in recommender systems: a survey," ACM Computing Surveys, vol. 55, pp. 1-37, 2020.

[18] W. Fan et al., "Graph neural networks for social recommendation." In The world wide web conference, pp. 417-426. 2019.

[19] Q. Cao, H. Shen, J. Gao, B. Wei, and X. Cheng, "Popularity prediction on social platforms with coupled graph neural networks," Proceedings of the 13th international conference on web search and data mining, pp. 70-78. 2020.

[20] M. Wang et al., "Deep graph library: A graph-centric, highly-performant package for graph neural networks," arXiv preprint arXiv:1909.01315, 2019.

[21] A. Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," In Advances in neural information processing systems, pp. 8024-8035. 2019.

[22] W. Falcon et al., "Pytorch lightning," GitHub 3, 2019.

[23] M. Honnibal and I. Montani, "spaCy: Industrial-Strength Natural Language Processing in Python," available at https://spacy.io, 2021.

[24] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," The journal of machine learning research 15, no. 1, pp. 1929-1958, 2014.

[25] R. Prasad et al., "The penn discourse treebank 2.0 annotation manual," December 17, 2007.

[26] J. Li, E. Durmus, and C. Cardie, "Exploring the role of argument structure in online debate persuasion," Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pp. 8905–8912, 2020.

[27] J. Zeng et al., "What changed your mind: The roles of dynamic topics and discourse in argumentation process," Proceedings of The Web Conference, pp. 1502-1513, 2020.

[28] L. Wu et al., "Graph neural networks for natural language processing: A survey," Foundations and Trends® in Machine Learning 16.2, pp. 119-328, 2023.

[29] T. Ji, W. Yuanbin, and L. Man, "Graph-based dependency parsing with graph neural networks," Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Italy, pp. 2475-2485, 2019.

[30] B. Liang, S. Hang, G. Lin, C. Erik, and X. Ruifeng, "Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks," Knowledge-Based Systems, vol. 235, p. 10764, 2021.

[31] R. Li et al., "Dual graph convolutional networks for aspect-based sentiment analysis," Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pp. 6319–6329, 2021.

[32] L. Song, Y. Zhang, Z. Wang, and D. Gildea, "A graph-to-sequence model for AMR-to-text generation," Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Long Papers, Australia, pp. 1616-1626, 2018.

[33] Y. Chen, L. Wu, and M. J. Zaki, "Graphflow: Exploiting conversation flow with graph neural networks for conversational machine comprehension," Proceedings of the 29th International Joint Conference on Artificial Intelligence, Japan, pp. 1230-1236, 2020.

[34] V. Niculae, J. Park, and C. Cardie, "Argument mining with structured SVMs and RNNs," Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada, pp. 985-995, 2017.

[35] C. Tan, V. Niculae, C. Danescu-Niculescu-Mizil, and L. Lee, "Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions," Proceedings of the 25th International Conference on World Wide Web, pp. 613-624, 2016.

[36] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," Proceedings of the 27th International Joint Conference on Artificial Intelligence, pp. 3634-3640, 2018.

[37] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," Proceedings of the AAAI conference on artificial intelligence, vol. 33, no. 01, pp. 922-929, 2019.

[38] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 12026-12035, 2019.

[39] https://github.com/quanmai/SGA.