Hierarchical Neural Network: Integrate Divide-and-Conquer and Unified Approach for Argument Unit Recognition and Classification

Yujie Fu[a], Suge Wang[a,b,*], Xiaoli Li[c], Deyu Li[a,b], Yang Li[d], Jian Liao[a], Jianxing Zheng[a]

[a]*School of Computer and Information Technology, Shanxi University, China*
[b]*Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, China*
[c]*Institute for Infocomm Research, A*STAR, Singapore*
[d]*School of Finance, Shanxi University of Finance and Economics, China*

Abstract

Argument unit recognition and classification (AURC) is a promising and critical research topic in argument mining, which aims to extract the argument units that express support or opposing stance in a given argumentative text under controversial topics. Existing studies treated the AURC as a sequence labeling problem and designed a unified approach to predict argument unit boundary and argument unit stance simultaneously. In this paper, we propose a general framework **h**ierarchical **n**eural **n**etwork (HNN) for AURC, by fusing two different approach: divide-and-conquer approach and unified approach. The divide-and-conquer approach considers the correlation of the two tasks inherent in AURC (task 1: argument unit recognition, AUR and task 2: argument unit classification, AUC), and jointly optimize them for prediction by a novel probability transition matrix. Finally, we used a token-level attention mechanism to efficiently fuse probability distributions obtained by our proposed divide-and-conquer approach and existing unified approach. Experimental results on two benchmark datasets demonstrate the effectiveness of our proposed framework.

Keywords: Argument Mining; Argument Unit Recognition and Classification; Sequence Labeling; Stance Detection;

*Corresponding author. Email: wsg@sxu.edu.cn

1. Introduction

The act or process of giving reasons to support or opposing a viewpoint in order to persuade a known or unknown audience is called argumentation [26]. Argumentation are fundamental human skills that play an important role in education, everyday conversation, and many professional settings including journalism, politics, and law [42]. With the rapid development of Internet technology and social media, users will generate a large amount of subjective data such as opinions and comments on a controversial topic. The research on these subjective data contains huge commercial and academic value. The purpose of argument mining is to study how to automatically identify arguments and extract argument relationships from subjective data, so as to meet people's higher demand for information retrieval and information extraction [35].

Most existing methods perform argument mining at the *discourse*-level, such as argument unit recognition (AUR) [2, 16], argument unit classification (AUC) [28, 34], and argument relationship detection (ARD) [35, 9]. These discourse-level methods address the identification of argument structures within a single document, but they do not take into account the relevance of externally defined controversial topics. Discourse-level argument mining models are highly dependent on the text types (such as scientific publications [20] and persuasive essays [36]) for which they were designed and do not work well when applied to other text types [7]. Therefore, another branch of argument mining, information-seeking argument mining, was proposed. Unlike discourse-level argument mining, information-seeking argument mining aims to identify argumentative sentences relevant to a given topic. The goal of information-seeking argument mining is to identify broad and diverse argument units that reflect different viewpoints on a controversial topic [42].

In particular, the combination of the first two tasks, i.e., argument unit recognition and classification (AURC) [42], aiming to extract the arguments of a stance expression from a given text under a controversial topic, has gained increasing attention recently. Stab et al. [37] and Fromm et al. [12] formulate AURC as a *sentence*-level classification task that attempts to determine whether each sentence in a given document is a non-argument, supporting argument, or opposing argument. But this solution may not be enough to solve AURC problem, as shown in the two examples below.

Example 1: Myth: [Having an abortion will help our relationship by removing the stress of a pregnancy]$_{\text{pro}}$. [topic = *abortion*]

Example 2: [Nuclear energy may have horrific consequences if an accident

occurs]$_{\text{con}}$, but [it has an enormous capacity for energy production with no carbon emissions]$_{\text{pro}}$. [topic = *unclear energy*]

In the above two examples, both Example 1 and 2 are classified as supporting argument class, because it contain argument units with pro stance. Particularly, in Example 1, the part underlined is *pro argument unit*, indicating authors favor the topic of abortion due to the reason 'abortion will help our relationship by removing the stress of a pregnancy'. In Example 2, the part with a wavy line is *con argument unit*, indicating authors oppose the topic if an accident occurs. However, interestingly, Example 2 also contains *pro argument unit* due to the benefits of enormous energy production without carbon emissions.

We observe determining whether a sentence containing the argument will not be sufficient for argument extraction. Example 1 is an argumentative sentence, but only the underlined argument unit is the core of the argument, as it describes more precisely why the author holds this stance. In addition, from Example 2, there may be more than one arguments unit with different stances in the same sentence. Hence, the sentence-level argument unit recognition task cannot identify the arguments corresponding to different stances mentioned in the given sentence. Such gap motivates us pinpoint more precise or fine-grained span-level or token-level argument expressions which can convey specific reasons of different stances (pro and con) within a given text.

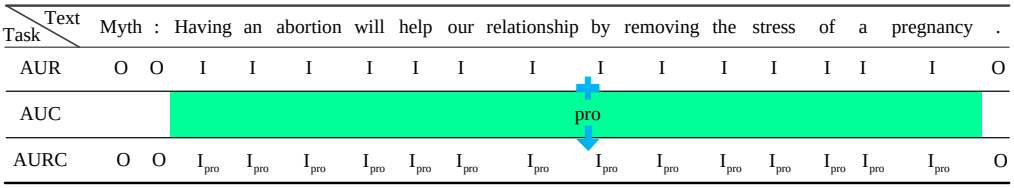

Figure 1: Different tasks involved in AURC

Therefore, Trautmann et al. [42] has provided a benchmark datasets and used bidirectional encoder representation from transformers (BERT) model to address the AURC problem. In addition, Trautmann et al. [42] formulated the AURC as a sequence labeling task where the argument unit labels (i.e., non-argument and argument) and stance labels (i.e., pro and con) are mapped into a unified space (see AURC in Figure 1), so that AUR and AUC are completed at the same time. As shown in Figure 1, from the unified label space, it can obtain both argument unit boundary that indicates whether each token belongs to an argument unit or not (O indicates corresponding token is outside or not part of the argument unit, while

I denotes the token is inside or part of the argument unit) and stance class of the argument unit (pro). Here, the unified label $\mathrm{I}_{\mathrm{pro}}$ consists of both boundary (I) and stance class (pro). Essentially, it is a unified approach by performing three-class classification for each token, with three class labels: O, $\mathrm{I}_{\mathrm{pro}}$, $\mathrm{I}_{\mathrm{con}}$.

From the perspective of sequence labeling task, AURC is similar to named entity recognition (NER) task, both are labeling specific content in a given text. But the length of the argument unit in AURC is significantly longer than that of the entity type in NER. In addition, each argument unit contains complete semantics and can exist as a separate clause. Obviously, identifying boundaries of argument unit and classifying stance of argument unit are two difficult problems of this task. Therefore, using the identified boundaries of argument unit to understand the complete semantics of the argument unit, so as to classify the stance of the argument unit under a controversial topic is an idea to solve the AURC task.

We address token-level AURC task by proposing **h**ierarchical **n**eural **n**etwork (HNN) framework, which is able to effectively integrate divide-and-conquer approach and unified approach, to improve the discrimination power of the framework for AURC task. More specifically, we propose a *divide-and-conquer approach* to tackle the AURC task, where we focus on two inherent underlying subtasks, AUR (boundary with labels O and I) and AUC (stance labels could be pro and con). Both subtasks are binary classification and thus relatively simple. Divide-and-conquer approach can potentially generate better prediction results as we focus on them individually. In addition, to alleviate the error cascading problem brought by the divide-and-conquer approach, we jointly optimize the two subtasks together. In particular, we first perform AUR prediction to identify argument unit spans, and subsequently perform AUC to identify corresponding stance class for each argument unit span, and construct a probability transition matrix, which is used to transfer the probability from the boundary space to the unified space. In addition, we leverage the results of divide-and-conquer approach and existing unified approach to construct a unified encoding layer to simultaneously recognize argument units and classify their corresponding stances. Finally, we integrate the label probabilities obtained by the above two approach at token-level to obtain the final unified labels. The main contributions of our work are three-fold:

- We propose a novel and generic sequence-to-sequence based hierarchical neural network framework that integrates *divide-and-conquer approach* and *unified approach* effectively for the token-level AURC task.
- In order to better integrate two probability distributions in unified space obtained by two different approach, we design a *token-level attention* method.

- Extensive experimental results show that the proposed framework performs significantly better than state-of-the-art methods on two benchmark datasets.

The remainder of this paper is organized as follows. Section 2 introduces the related work. Section 3 details the proposed HNN. Section 4 discusses experiment and analysis. Section 5 presents the conclusions of the study.

2. Related Work

In recent years, deep learning methods have been widely used in different fields, across real-life applications or theoretical research [3, 10, 5, 1, 44, 32]. Among them, in natural language processing domain, the earliest research on argument mining began in some specific application domains, such as legal documents, online reviews and debates [25, 4, 27]. In recent years, however, due to the development of deep learning technology and abundance of annotated corpora, argument mining has gradually prospered across different domains. In existing work [12, 37], some researchers addressed the problem of topic-focused argument extraction on the sentence-level.

In general, argument mining mainly focuses on the microstructure of argumentation, and its representative models mainly include: claim-premise model, Toulmin model [41], standard approach [40], Freeman model [11], argumentation scheme [43].

Most argument mining models [45, 33] rely on the claim-premise model (where each argument must have an argumentative structure, i.e., consist of both claim and associated supported premise), which is, however, hardly applicable to regular texts that do not contain an explicit argumentative structure, e.g., social media data [17]. Information-seeking argument mining was therefore proposed. It solves the following task: given a *controversial claim* or *topic* (e.g., abortion), we detect pro or con statements from some relevant texts. In this context, an argument is usually defined as a short text or span, that provides stance evidence or reasoning about a topic, e.g., favor or oppose the topic [37].

AUR is the first step in argument mining. Some researchers treat it as a sentence classification task. For instance, Moens et al. [27] first extracted different features of sentences involving lexical, syntactic, semantic, and discourse properties, and subsequently trained a multinomial Naïve Bayes classifier and maximum entropy model, so that they can classify a test sentence into a argumentative and non-argumentative sentence. Under the claim-premise model, Li et al. [22] regarded AUR as a sequence labeling problem and trained a recurrent neural

network model to more precisely detect argument unit boundaries. Ajjour et al. [2] further discussed the effectiveness of different features in token-level AUR. However, they ignored the stance class (pro or con) of argument units. Peldszus and Stede [29] used discourse analysis as the starting point to classify the argument units, but they ignored the topic-dependency (a controversial claim) in information-seeking argument mining.

As AUR is often transformed into a sequence labeling problem, conditional random field (CRF) is a commonly used method [16, 31]. Because the context of the argumentative text has strong semantic relevance, some scholars have begun to use neural networks (recurrent neural network, long short-term memory networks) combined with CRF to identify the boundaries of argument units [22, 30].

Recently, Trautmann et al. [42] combined argument unit recognition and classification tasks, and created a new token-level (fine-grained) benchmark corpus. Their motivation was that token-level models support more specific selection of argumentative spans within sentences. They have used BERT model to directly conduct three-class classification so that they can predict boundary and stance simultaneously.

In this paper, different from all the existing work, we propose a divide-and-conquer approach to tackle AUR subtask and AUC subtask individually and then assemble them to make prediction. In addition, since divide-and-conquer approach and unified approach make predictions from different and complementary perspectives, integrating them could potentially lead to more accurate predictions. Therefore, we also design a token-level attention component to integrate them to further boost their performance for AURC task.

3. Framework

This paper proposes a hierarchical neural network (HNN) framework to effectively integrate the new *divide-and-conquer approach* and existing *unified approach* for AURC task prediction. In particular, the divide-and-conquer approach decouples the overall AURC task into two inherent subtasks and tackles them in sequence, and then re-integrate their learnt probability distribution knowledge at the higher level for accurate prediction.

Specifically, the proposed HNN framework jointly optimizes three tasks: AUR, AUC, and AURC. On the basis of the AUR module (bottom block), we design AUC module (middle block) to construct a probability transition matrix, which derives a boundary probability distribution to a unified label space. The AUR and AUC combined together for joint optimization is *divide-and-conquer approach*

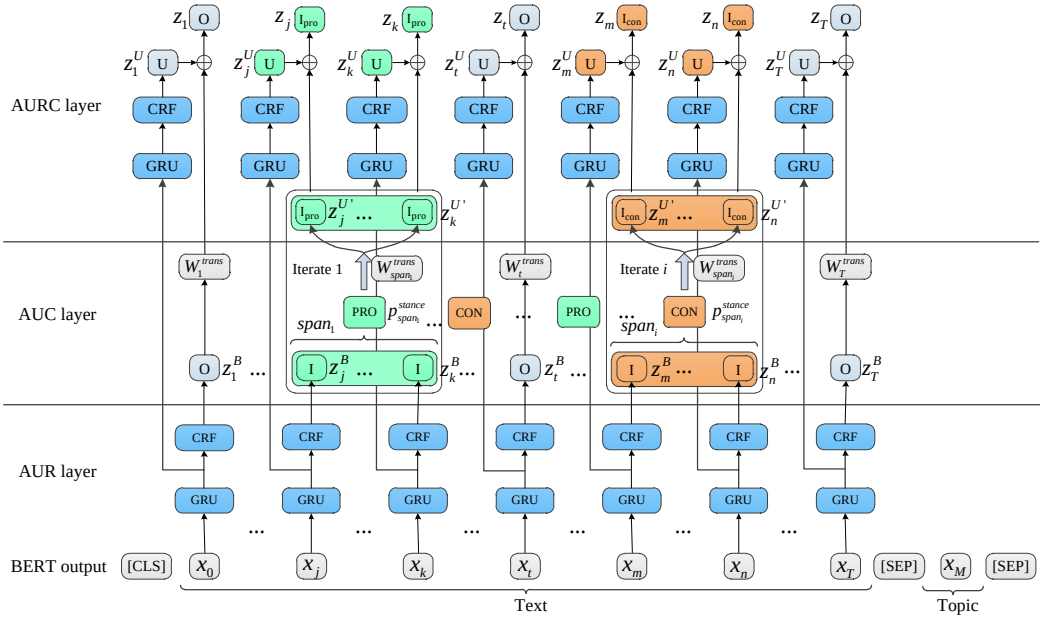

Figure 2: The framework of hierarchical neural network.

part. In addition, via existing unified approach and the word representation with boundary information, we can also generate a probability distribution in a unified sequence labeling space direct. It's *unified approach part*. Finally, the two probability distributions obtained by these two different approach are finally fused by a token-level attention (top block). The overall architecture of the proposed HNN framework is shown in Figure 2.

3.1. Task Definition

Following Trautmann et al. [42], we formulate the complete token-level AURC task as a sequence labeling problem and employ a unified tagging scheme $y^U = \{\mathrm{O}, \mathrm{I}_{\mathrm{pro}}, \mathrm{I}_{\mathrm{con}}\}$. In particular, O indicates that the corresponding token is not part of the argument unit (outside boundary), while each tag in $\{\mathrm{I}_{\mathrm{pro}}, \mathrm{I}_{\mathrm{con}}\}$ contains two parts of tagging information: the inside boundary of the argument unit, and the corresponding stance. For example, $\mathrm{I}_{\mathrm{pro}}$ denotes the token is part of *favoring* argument mention, while $\mathrm{I}_{\mathrm{con}}$ denotes the token is part of *opposing* argument mention. For a given input sequence $W^{text} = \{w_1^{text}, ..., w_t^{text}\}$ with length t, our goal is to predict a tag sequence $Y^U = \{Y_1^U, ..., Y_t^U\}$, where $Y_i^U \in Y^U$. The proposed HNN framework consists of three layers: AUR layer, AUC layer, and AURC layer. Note the HNN framework is a general sequence labeling framework

based on sequence-to-sequence, meaning different methods can be used for each of the above three layers. Here we take a sequence label model conditional random field based on bidirectional gate recurrent unit (BiGRU+CRF) as an example to introduce the working methods for the proposed framework, although other working methods may also be applied as we will show in experimental section.

The BERT is a pre-training model based on a bidirectional transformer, trained on a large amount of unlabeled data. Depending on the specific downstream task, the model can be initialized with pre-trained parameters, and then fine-tuned on the labeled dataset for that task. In recent years, significant amount of research have used BERT as an initialization method for word vectors as it has strong contextualized representation ability [24, 21, 13, 6, 14].

We employ BERT as the backbone of the framework as it has strong contextualized representation ability. The text and its associated topic words (e.g., *abortion* is a topic to indicate the domain of the text) are concatenated by forming a combined sequence as the input to BERT: [CLS], $w_1^{text}, ..., w_T^{text}$, [SEP], $w_1^{topic}, ..., w_M^{topic}$, [SEP], where [CLS] and [SEP] are special tokens. Then, BERT receives the sequence and outputs the representation of each token in the combined sequence. Due to the structural properties of the BERT, topic information is incorporated into the output contextual representation during the encoding process. The contextualized representations of each token $X = [x_1, x_2, ..., x_T]$ can be given as:

$$x_t = BERT(w_t^{text}), 1 \leq t \leq T \tag{1}$$

AUR layer. Based on the BERT encoding, we first use GRU^B to learn long term dependencies of tokens and encode the boundary information of its argument unit, where the boundary information can be used as a clue and indicator for unified label prediction. In particular, if the boundary label of the current token is I, it indicates that the word belongs to an argument unit. As such, the corresponding unified labels can only be $\mathrm{I}_{\mathrm{pro}}$, $\mathrm{I}_{\mathrm{con}}$, depending on its different stance class. Note the valid label set Y^B for argument boundary prediction is $\{\mathrm{I}, \mathrm{O}\}$, and the valid label set Y^U for unified label prediction is $\{\mathrm{I}_{\mathrm{pro}}, \mathrm{I}_{\mathrm{con}}, \mathrm{O}\}$.

The hidden representations h_t^B at the t-th time step $t \in [1, T]$ of the first GRU^B are concatenate as follows:

$$h_t^B = [\overrightarrow{GRU}^B(x_t); \overleftarrow{GRU}^B(x_t)] \tag{2}$$

where $[\cdot;\cdot]$ is the concatenation operator of two vectors.

Finally, the probability scores z_t^B for all tokens $x_i \in X$ over the boundary tags $\{\mathrm{I}, \mathrm{O}\}$ are calculated by a CRF layer:

$$z_t^B = p(y_t^B | h_t^B) = CRF(W^B h_t^B + b^B) \tag{3}$$

AUC layer. In the second layer of the HNN framework, we try to classify the stance of each potential argument unit where each token in corresponding span has label I. During training process of this layer, we input the real boundary labels (O and I), and judge the stance of each argument unit span ($\mathrm{I_{pro}}, \mathrm{I_{con}}$) separately. During testing process, however, we take the output of the predicted boundary detection of the first layer as the input to the second layer, as we do not have ground truth argument boundary for test examples. Specifically, we leverage a binary mask matrix to mask content outside the argument unit span as they are not part of the argument unit.

$$mask_t = \begin{cases} 0, & y_t^B = \mathrm{O} \\ 1, & y_t^B = \mathrm{I} \end{cases} \tag{4}$$

We repeatedly concatenate $mask^T (mask = [mask_1, mask_2, ..., mask_t])$ for d_m times, and then perform element-wise multiplication with the context representation X to obtain the mask representation X^m. The specific operation is shown in the following formula:

$$X^m = X \odot (mask^T \otimes d_m) \tag{5}$$

where $\otimes$ is a operator that repeatedly concatenates $mask^T$ for d_m times; d_m is the dimension of x_t.

Through the formula (5), we can get the context representation after mask. Since BERT is a continuous contextual representation, brute force truncation relying on the mask matrix may affect the semantics of word embedding representations. Thus, we utilize a GRU^M to fine-tune the semantic encoding h_t^m.

$$h_t^m = [\overrightarrow{GRU}^M(x_t^m); \overleftarrow{GRU}^M(x_t^m)] \tag{6}$$

Note topic-specific attention mechanisms has been shown to be beneficial for stance detection [45, 33, 39]. Furthermore, different tokens play different roles in an argument. To emphasize the important information composition in the argument span, we use a bilinear attention mechanism to calculate an attention score for each word in the argument unit span, where the topic vector q^t is a query. In formula (7) we provide four different methods for calculating the attention scoring function. The topic vectors here are generated by random initialization, and fine-tuned gradually as the training progresses. Then through a softmax layer, the normalized weights of the tokens can be obtained. Finally, we calculated the mask representation of the current argument span by weighted sum.

$$score(q^t; h_t^m) = \begin{cases} h_t^{mT} q^t, & \text{dot} \\ h_t^{mT} W q^t, & \text{general} \\ V^T \tanh(W h_t^m + U q^t), & \text{perceptron} \\ V^T \tanh(W(h_t^m; q^t) + b), & \text{bilinear} \end{cases} \tag{7}$$

$$\alpha_t = \frac{exp(score(q^t; h_t^m))}{\sum_{i=1}^{t} exp(score(q^t; h_t^m))} \tag{8}$$

$$H_{span_i}^{stance} = \sum_{t=1}^{T} \alpha_t h_t^m \tag{9}$$

The probability score $p_{span_i}^{stance}$ on the stance label of the current span i is calculated by a fully connected softmax layer:

$$p_{span_i}^{stance} = softmax(W^S H_{span_i}^{stance} + b^S) \tag{10}$$

Inspired by Li et al. [23], we construct a probability transition matrix based on the probability $p_{span_i}^{stance}$. However, different from Li et al. [23], we utilize prior knowledge in transition probabilities between boundary tags and unified tags. One of the reasons is that in the AURC task, the argument unit is usually relatively long and contains complete semantics to provide argument evidence. Establishing the transition probability for each argument unit span can effectively utilize the AUC classification results enhanced by topic information. In particular, a transfer vector W_t^T is constructed as follows:

$$W_t^T = \begin{cases} [1, 0, 0] & , y_t^B = O, \\ \left[0, p_{span_i}^{pro}, p_{span_i}^{con}\right] & , y_t^B = span_{(i,j)}. \end{cases} \tag{11}$$

where $span_{(i,j)}$ is the j-th token in $span_i$. When $y_t^B = \mathrm{O}$, it indicates that the current token does not belong to part of the argument unit, i.e., the probability of non-argument or non-class is 1. Correspondingly, the unified label after the transfer should be O, and the corresponding transfer vector is $[1, 0, 0]$. The three-dimensions correspond to the probabilities of the three stance labels. On the other hand, when y_t^B belongs to a certain $span_i$, the stance label probability $\left[p_{span_i}^{non}, p_{span_i}^{pro}, p_{span_i}^{con}\right]$ of $span_i$ predicted by the second layer is used to assign the transition probability of each word in the $span_i$, where $p_{span_i}^{non}$=0 as current token is part of the argument unit.

Likewise, on the basis of the AUR results, we map the probability scores of boundary tags to unified tags through a transition matrix. Finally, we get the label mapping result $z_t^{U'}$ based on the transition matrix.

$$z_t^{U'} = z_t^B W^T \tag{12}$$

The boundary probability z_t^B of the argument unit span can be combined with the transition matrix W^T to determine its distribution in the unified labeling decision space $z_t^{U'}$.

AURC layer. The hidden representation h_t^U used to predict the unified label is calculated as follows:

$$h_t^U = [\overrightarrow{GRU}^U(h_t^B); \overleftarrow{GRU}^U(h_t^B)] \tag{13}$$

Similarly, the scores over the unified tags z_t^U are obtained as below:

$$z_t^U = p(y_t^U | x_t) = CRF(W^U h_t^U + b^U) \tag{14}$$

At this point, we obtained two probability distribution of the unified label space from two different and complementary methods, namely, 1) the probability distribution $z_t^{U'}$ is obtained by divide-and-conquer approach part with two subtasks in AUC layer and AUR layer, and 2) the unified label distribution z_t^U is obtained from the unified approach part which makes a prediction in the unified label space. To effectively fuse the information from these two label probabilities, we design a token-level attention mechanism, where we assign fusion weights at the *token-level* to the two probability distributions for each token.

$$s_t = V^T \tanh(W[z_t^{U'^T} \| z_t^{U^T}] + b) \tag{15}$$

where $[\cdot\|\cdot]$ means that two matrixs are concatenated by column. s_t is the score at t-th time step. Finally, a score vector with size 1×2 is obtained.

Finally, the normalized weight β_t of the score at step t is calculated by a softmax layer. Finally, the distribution z_t of the final unified label space is calculated by formula (17).

$$\beta_t = softmax(s_t) \tag{16}$$

$$z_t = \beta_t \begin{bmatrix} z_t^{U'} \\ z_t^U \end{bmatrix} \tag{17}$$

3.2. Loss

In general, the framework consists of three parts: 1) the first part detects the boundaries of argument unit spans from a given text, 2) the second part performs the stance detection for all the spans of argument units, 3) the third part completes the information fusion of divide-and-conquer approach and unified approach. Among them, the two tasks, AUR and AUC, are independent but jointly optimized. Correspondingly, the overall loss of the framework also contains 3 parts.

First part: we aim to minimize the negative log-likelihood in CRF.

$$L^B = -\frac{1}{T}\sum_{t=1}^{T} y_t^B \log(z_t^B) \tag{18}$$

The gradients with respect to the parameters can be calculated efficiently through the forward-backward algorithm in the CRF layer and back propagation in the neural networks.

The second part loss is cross-entropy error:

$$L_{span}^{stance} = -\sum_{i=1}^{n}\sum_{j=1}^{m} y_{span_i}^{stanc_j} \log(\hat{p}_{span_i}^{stance_j}) \tag{19}$$

where n is the number of spans and m is the number of stance classes.

Similar to the first part, the third part uses the loss of argument unit recognition and unified label after fusion to further constrain the argument unit recognition task.

$$L^{Joint} = -\frac{1}{T}\sum_{t=1}^{T} y_t^B \log(z_t^B) - \frac{1}{T}\sum_{t=1}^{T} y_t^U \log(z_t) \tag{20}$$

The final loss function is defined as follows:

$$L = L^B + L_{span}^{stance} + L^{Joint} \tag{21}$$

4. Experiments

In this section, we compare the proposed HNN framework with existing state-of-the-art models using benchmark AURC-8 dataset [42] and SECA dataset [24].

4.1. Datasets

AURC-8 [1]. We employ the latest AURC-8 dataset published in 2020 which includes 8 topics, annotating with the token-level argument units and stances [42]. The topic names have been used together with text for argument boundary detection introduced in Section 3. In particular, each topic contains 1000 sentences. #arg-sent, #non-arg, #arg-unit represents the number of argument sentences, non-argument sentences, and number of argument units. Each argument sentences contains at least one argument span and each argument units can be classified two argument unit categories: PRO and CON. The statistics of the 8 datasets are shown in the Table 1.

SECA [2]. Emotion cause analysis (ECA) aims to identify the reasons behind a certain emotion expressed in the text, but such *clause*-level ECA (CECA) can be ambiguous and imprecise. As such, Li et al. [24] proposed *span*-level ECA (SECA), and manually annotated cause spans based on ECA [15] dataset. This task is similar to AURC, so we also evaluate the framework on this dataset. In particular, each context in this task contains an emotion expression (or emotion category) and one cause span. In addition, each cause span can be classified into one of seven emotion categories, including anger, disgust, fear, happy, sad, shame, and surprise. The **SECA** dataset contains 820 emotion-cause instances (each instance contains one span only) and 1594 no-cause instances. Table 2 shows the distribution of these emotions in the dataset.

Table 1: The proportion of AURC-8.

		number	#arg-sent	#non-arg	#arg-unit
T1	abortion	1000	424	576	458
T2	cloning	1000	353	647	380
T3	marijuana legalization	1000	630	370	689
T4	minimum wage	1000	630	370	703
T5	nuclear energy	1000	623	377	684
T6	death penalty	1000	598	402	651
T7	gun control	1000	529	471	587
T8	school uniforms	1000	713	287	821
	total	8000	4500	3500	4973

[1] https://github.com/trtm/AURC

[2] https://github.com/xxxyyy2020/boundary-master

Table 2: The proportion of SECA.

	emotion-cause	no-cause	total
anger	199	284	483
disgust	38	57	95
fear	144	279	423
happy	211	268	479
sad	107	468	575
shame	68	78	146
surprise	53	160	213
total	820	1594	2414

4.2. The experimental setup

Data Split: In **AURC-8**, Trautmann et al. [42] presented two different dataset splits: 1) an in-domain split (4000 / 800 / 2000 for train / dev / test) and 2) a cross-domain split (4200 / 600 / 1200). In the cross-domain setup, Trautmann et al. [42] defined topics T1-T5 to be in the train set, topic T6 in the dev set and topics T7 and T8 in the test set. We will evaluate different methods using both splits. In **SECA**, however, the training set and test set are not divided. Therefore, we perform standard five-fold cross-validation on the dataset to evaluate the performance of the proposed framework, following Li et al. [24].

Evaluation Metrics: To evaluate the performance of different models at a more fine-grained level (token-level), we employ the macro-averaged F1 score (or macro F1 score), which is computed by taking the arithmetic mean (aka unweighted mean) of all the per-class F1 scores. In other words, we compute the mean value of F1 for the three classes in set $y^U = \{O, I_{pro}, I_{con}\}$ [42], where F1 for each class is computed for all tokens in the evaluation set and the mean F1 score is then finally reported. In addition, we also show the results of the HNN framework in **p**recision (P) and **r**ecall (R).

Compared Models: We compare the HNN framework with the following methods:

- **BERT**: Trautmann et al. [42] uses BERT [8] for AURC task, which is a state-of-the-art pretrained model that achieves impressive results on many NLP tasks, including sequence labeling. BERT has been tested and extended in Trautmann et al. [42].

- **Li-unified** [23] is a seqence labelling method, aiming to detect the opinion

targets and predict the sentiment polarities over the opinion targets in target-based sentiment analysis (TBSA) task.

- **BiGRU-CRF** [19] is a bidirectional gated recurrent unit model with CRF decoding layer, a commonly used sequence labeling model.
- **IDCNN-CRF** [38] is a faster alternative to bidirectional gated recurrent unit for NER, which has better capacity than traditional convolutional neural networks for large context and structured prediction.
- **CNN-NER** [46] used a convolutional neural network to model the interaction between adjacent entity spans with special correlations.
- **W^{2}NER** [21] proposed a novel alternative to NER by modeling NER as word-word relation classification. This architecture solves the bottleneck of NER by effectively modeling the adjacent relations between entity words.

Other Settings: We first use pre-trained BERT-base-uncased to encode the AURC-8 and SECA datasets, then follow the default settings of BERT for fine-tuning. Adam [18] optimizer is used with learning rate 1e-5. The random seed is 2021. The dimension of the topic vector in AUC is 300. The training batch size is 32. We fine-tuned for 10 epochs in the AURC task and for 15 epochs in the SECA. Maximum sentence length is 90, dropout is 0.3. The gate dimension of Li-unified is 300*300. The hidden representation dimension of BiGRU is 150, and the number of layers is 1. The filters of IDCNN-CRF is 300. The code has been made publically available online to advance science in this area and faciliates researchers for their model comparions and further new model development [3].

4.3. Experimental results and analysis

4.3.1. Comparison results of different methods for AURC-8.

We show the comparison performance of the various methods in Table 3. As mentioned, Trautmann et al. [42] adopted BERT for the AURC task, and its performance is directly taken from the paper. *-HNN means to use the HNN framework on the basis of *, that is, use different working methods of * for AUR and unified approach part prediction. Among them, the AUC task adopts BiGRU+attention model.

[3] https://github.com/cmfyj/Argument-Unit-Recognition-and-Classification

Table 3: Performance comparison of different methods in AURC-8.

		in-domain			cross-domain		
	Model	P	R	F1	P	R	F1
span-based	CNN-NER [46]	0.638	0.602	0.617	0.554	0.474	0.482
	W^2NER [21]	0.589	0.605	0.596	0.597	0.534	0.558
sequence to sequence	BERT [42]	-	-	0.654*	-	-	0.563*
	BERT-Li-unified [23]	0.641	0.641	0.641	0.566	0.539	0.547
	BERT-Li-unified-HNN	0.683	0.669	0.676	0.606	0.577	0.580
	BERT-BiGRU-CRF [19]	0.657	0.668	0.662	0.595	0.558	0.569
	BERT-BiGRU-CRF-HNN	**0.685**	0.681	**0.683**	0.601	0.577	0.585
	BERT-IDCNN-CRF [38]	0.649	0.670	0.658	0.584	0.554	0.563
	BERT-IDCNN-CRF-HNN	0.679	**0.683**	0.681	0.612	**0.584**	0.584

From Table 3, we observe that among the two different types of methods, these span-based methods perform the worst. As they need to generate the start and end positions of spans for different lengths, the longer the span length, the more difficult it is to accurately capture the semantics contained in the span, and the more ambiguous the positioning of start and end positions. In the sequence-to-sequence based approach, we conduct experiments by adding the HNN framework to different base models. The transition matrix in Li et al. [23] adopts the same probability for different stance, and its output layer uses the sentiment consistency component to control the consistency of the label sequence. For AURC task, its performance is not ideal. However, the performance of Li-unified-HNN has been significantly improved after leveraging the proposed HNN framework. In fact, by adding the HNN framework to three different models, namely Li-unified, BiGRU-CRF, and IDCNN-CRF, we observe that Li-unified-HNN, BiGRU-CRF-HNN, and IDCNN-CRF-HNN have all been improved consistently in terms of F1 score by 0.35, 0.21, and 0.23 for in-domain scenario and by 0.33, 0.16, and 0.21 for cross-domain scenario respectively. In contrast, cross-domain for AURC is obviously more difficult and challenging.

4.3.2. Comparison results of different methods for SECA.

In Li et al. [24], the original SECA (ori-SECA) task is defined as: given a sentence and a emotion category, extract the corresponding emotional cause span. However, in order to be consistent with the setting of the AURC task, according to the given sentence, we need to extract the cause span and judge its corresponding emotion category, namely variant SECA (var-SECA). We show the comparison performance of SECA in Table 4, where the experiments of the ori-SECA part

were selected from Li et al. [24].

Table 4: Performance comparison of different methods in SECA.

Input	Model	P	R	F1
sentence emotion category (ori-SECA)	Ghazi et al. [15]	0.666	0.593	0.628
	BERT+Softmax [24]	0.838	0.876	0.856
	BERT+GRU [24]	**0.883**	0.868	0.875
	BERT+CRF [24]	0.866	**0.890**	**0.878**
sentence (var-SECA)	BERT [42]	0.778	0.840	0.796±0.0357
	BERT-Li-unified [23]	0.819	0.854	0.823±0.0591
	BERT-Li-unified-HNN	0.849	0.903	**0.870±0.0282**
	BERT-BiGRU-CRF [19]	0.799	0.879	0.827±0.0298
	BERT-BiGRU-CRF-HNN	0.851	**0.905**	0.869±0.0386
	BERT-IDCNN-CRF [38]	0.791	0.857	0.816±0.0399
	BERT-IDCNN-CRF-HNN	**0.862**	0.880	0.864±0.0314

From the results in Table 4, it can be seen that after adding the HNN framework, the performance of the sequence labeling model is significantly improved (by about 0.53, 0.42, and 0.48, respectively). Comparing with the ori-SECA task with a given emotion category, the F1 value is only 0.008-0.014 lower than the former best results, even though the corresponding emotion category is not given in var-SECA task. Clearly the var-SECA task presented in this paper is obviously more challenging. Without a given emotion category, the performance of the two tasks is still somewhat comparable. It should be noted that the former only conducts experiments on 820 pieces of data with emotion cause (only including instances with emotion category cause), while this research conducts experiments on all 2414 pieces of data (including instances without emotion reasons span) experiment. It is worth mentioning that although topic information is not included in the SECA dataset, the proposed framework is highly competitive by considering long-range cause span information. Overall, on both AURC and SECA datasets, the consistent improvements against state-of-the-art models demonstrates the effectiveness and generality of the proposes HNN framework.

4.3.3. Ablation experiments of different approach for AURC-8.

In order to compare the impact of the divide-and-conquer and unified approach in the HNN framework on the performance, we designed the following ablation experiments for analysis, and the experimental results are shown in Table 5. [d&c]

means that AUR is performed first, followed by AUC, and the two tasks are optimized independently. [d&c part, -w/o unified] means to only use the divide-and-conquer approach part to complete the AUR and AUC in HNN, without using unified approach. The difference between the two contrasting methods is whether the two subtasks are optimized independently or jointly. In the HNN framework, the input of the unified part is the representation h_t^B with boundary information, [x_t for unified] means that the output x_t of BERT is used directly as the input for the unified part.

Table 5: Performance comparison of results without different components. The best results of each basic model are highlighted in bold.

	in-domain			cross-domain		
Model	P	R	F1	P	R	F1
BERT-Li-unified [unified]	0.641	0.641	0.641	0.566	0.539	0.547
BERT-Li-unified [d&c]	0.631	0.631	0.631	0.585	0.539	0.549
BERT-Li-unified-HNN [d&c part, -w/o unified]	0.680	0.667	0.673	0.604	0.576	0.580
BERT-Li-unified-HNN [x_t for unified]	0.670	0.660	0.662	**0.613**	0.562	0.574
BERT-Li-unified-HNN	**0.683**	**0.669**	**0.676**	0.606	**0.577**	**0.580**
BERT-BiGRU-CRF [unified]	0.657	0.668	0.662	0.595	0.558	0.569
BERT-BiGRU-CRF [d&c]	0.617	0.637	0.625	0.599	0.541	0.552
BERT-BiGRU-CRF-HNN [d&c part, -w/o unified]	**0.686**	0.681	0.682	0.603	**0.579**	0.582
BERT-BiGRU-CRF-HNN [x_t for unified]	0.671	0.679	0.675	**0.627**	0.568	0.579
BERT-BiGRU-CRF-HNN	0.686	**0.682**	**0.683**	0.601	0.577	**0.585**
BERT-IDCNN-CRF [unified]	0.649	0.670	0.658	0.584	0.554	0.563
BERT-IDCNN-CRF [d&c]	0.623	0.630	0.627	0.595	0.554	0.564
BERT-IDCNN-CRF-HNN [d&c part, -w/o unified]	0.676	0.679	0.677	0.607	0.580	0.580
BERT-IDCNN-CRF-HNN [x_t for unified]	0.659	**0.685**	0.670	0.596	0.565	0.574
BERT-IDCNN-CRF-HNN	**0.679**	0.683	**0.681**	**0.612**	**0.584**	**0.584**

In Table 5, we give a comparison of different working methods, from the results we can see that the fusion of different working methods is necessary. Compared to the unified method, the divide-and-conquer approach performs poorly in most model for different domain. Because there is error propagation between the two subtasks of the divide-and-conquer approach. And [d&c part] can significantly reduce the impact of error propagation by jointly optimizing the two subtasks and thus significantly improve the performance of the model. Moreover, from the comparison of (x_t for unified) and *-HNN results, it can be seen that the representation containing boundary information as input is more favorable for the prediction of unified labels. We notice *-HNN achieves the best results for both scenarios, as it can jointly optimize the two subtasks (AUR and AUC) and integrates the results of the two approachs (divide-and-conquer and unified). In

general, jointly optimizing these two subtasks, integrating boundary information for unified, and fusing two different methods to improve the HNN framework from different perspectives, have improved the performance of the proposed framework to varying degrees.

4.3.4. Effectiveness of different attention scoring functions.

In the AURC layer, we use a bilinear attention mechanism to fuse the information of different words. As shown in formula (7), the attention scores are calculated differently. To verify the performance of different scoring functions, we conduct comparative experiments on the AURC dataset. The experimental results are shown in Table 6.

Table 6: Performance comparison of results with different attention scoring functions. The best results of each basic model are highlighted in bold.

	in-domain			cross-domain		
Model	P	R	F1	P	R	F1
BERT-Li-unified-HNN [dot]	0.677	0.667	0.672	0.597	0.593	0.592
BERT-Li-unified-HNN [general]	0.682	0.666	0.674	0.603	0.574	0.583
BERT-Li-unified-HNN [perceptron]	0.674	0.663	0.665	**0.615**	**0.598**	**0.604**
BERT-Li-unified-HNN [bilinear]	**0.683**	**0.669**	**0.676**	0.606	0.577	0.580
BERT-BiGRU-CRF-HNN [dot]	0.654	0.667	0.660	0.604	**0.599**	**0.600**
BERT-BiGRU-CRF-HNN [general]	0.684	0.671	0.677	0.605	0.578	0.586
BERT-BiGRU-CRF-HNN [perceptron]	**0.706**	0.652	0.672	**0.611**	0.575	0.587
BERT-BiGRU-CRF-HNN [bilinear]	0.686	**0.682**	**0.683**	0.601	0.577	0.585
BERT-IDCNN-CRF-HNN [dot]	0.672	0.671	0.671	**0.615**	0.588	**0.595**
BERT-IDCNN-CRF-HNN [general]	**0.706**	0.671	**0.686**	0.598	0.572	0.572
BERT-IDCNN-CRF-HNN [perceptron]	0.672	0.675	0.673	0.599	**0.590**	0.593
BERT-IDCNN-CRF-HNN [bilinear]	0.679	**0.683**	0.681	0.612	0.584	0.584

According to the experimental results of Table 6, we observe that different attention scoring functions have a relatiely small impact on the performance of the HNN framework. During the experiments, we adopt the same hyperparameters for different frameworks built with different attention scoring functions. This is why there is no single scoring function that achieves optimal experimental results across all datasets. Moreover, the experimental results show that none of the four attention scoring functions has a significant performance advantage.

4.3.5. The influence of different training methods for the subtask of divide-and-conquer.

As mentioned in the previous, [d&c] methods are independent sequential method, [d&c parts] are jointly trained and optimized. In Table 7, we explore the differences brought about by the two training methods. [d&c step1] represents the first step in the independent sequential method for AUR, [d&c step2] means using the topic-specific attentional mechanism to complete AUC. Similarly, [d&c parts step1] represents the first layer in divide-and-conquer part of the framework, [d&c parts step2] represents the second layer.

Table 7: The results of each subtask. The best scores in each subtask are shown in bold.

		in-domain			cross-domain		
Task	Model	P	R	F1	P	R	F1
AUR	BERT-Li-unified [d&c step1]	0.769	0.770	0.770	0.744	0.714	0.716
	BERT-Li-unified-HNN [d&c parts step1]	**0.785**	0.779	0.781	0.761	0.751	0.753
	BERT-BiGRU-CRF [d&c step1]	0.766	0.776	0.769	0.751	0.714	0.715
	BERT-BiGRU-CRF-HNN [d&c parts step1]	0.782	**0.781**	**0.782**	**0.769**	0.755	0.758
	BERT-IDCNN-CRF [d&c step1]	0.766	0.770	0.768	0.756	0.731	0.734
	BERT-IDCNN-CRF-HNN [d&c parts step1]	0.777	0.779	0.778	**0.769**	**0.761**	**0.763**
AUC	[d&c step2]	0.788	0.786	0.785	**0.675**	**0.675**	**0.674**
	BERT-Li-unified-HNN [d&c parts step2]	0.789	0.789	0.789	0.655	0.651	0.645
	BERT-BiGRU-CRF-HNN [d&c parts step2]	**0.801**	**0.799**	**0.798**	0.648	0.644	0.644
	BERT-IDCNN-CRF-HNN [d&c parts step2]	0.791	0.790	0.790	0.672	0.666	0.658

From Table 7, we observe the divide-and-conquer part of the framework performs better as it can jointly optimize the two subtasks comparing with independent sequential method that optmizes the two subtasks in sequence, introducing propogation errors. Since framework aims to optimize the overall AURC task, each individual subtask may not be optimized individually, which leads to the subtask in the divide-and-conquer method performing the best in cross-domain AUC task. But apart from that, the HNN framework achieves optimal performance on other tasks. This is due to the fact that the two tasks share the underlying encoded representation, and the two tasks can be jointly optimized. The AUR task and the AUC task are simple, and the F1 is relatively high. The correlation between tasks can be constructed to a certain extent through the HNN framework. Therefore, the divide-and-conquer part exhibits a significant performance advantages. From the performance of in-domain and cross-domain, cross-domain is clearly more challenging. In cross-domain, both AUR and AUC are affected by topic absence. In particular, the performance of cross-domain is severely degraded, with the most dropping by 15.4%. Therefore, in order to improve the performance of AURC

tasks in cross-domain, the optimization of AUC deserves further research.

4.3.6. Comparison results of different components.

Since the HNN framework has two attention components: stance attention and token-level attention, in Figure 3, we present corresponding ablation experiments based on different sequence labeling methods to verify whether they have a positive impact on the whole framework. **-w/o token-level attention** means to remove the token-level attention components in the HNN framework and use a simple addition method instead. **-w/o topic attention** indicates that the topic-specific attention mechanism is not used in the AUC task, and the hidden layer representation at the last time in the BiGRU is used.

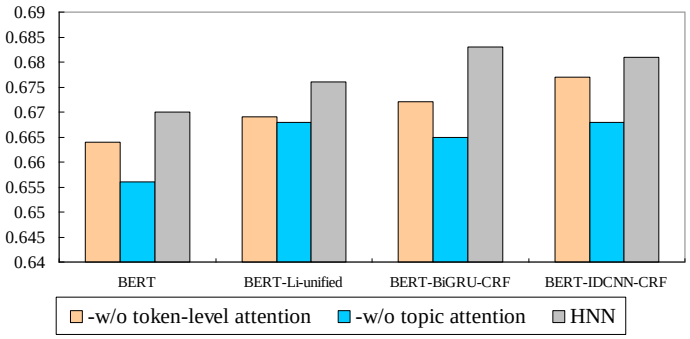

Figure 3: The results of three key components (in-domain).

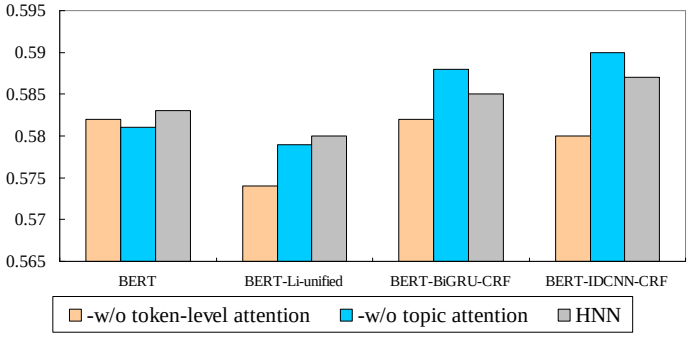

Figure 4: The results of three key components (cross-domain).

We observe from Figure 3 and Figure 4, in the in-domain, removing either component caused the performance of the HNN framework to degrade. In the cross-domain, since the topic of test set is unseen, removing the topic attention is

not significantly affected, which is confirmed with the results of the AUC task in Table 7. The results in Table 7 and Figure 4 show that, in the cross-domain, the performance of the AUC task is crucial and is an important factor affecting the AURC task. And token-level attention is particularly important in both domains. From the results of -w/o token-level attention, it can be seen that it is necessary to assign corresponding weights to each token in the information fusion. In the in-domain part, since the dataset contains topic information, topic attention is more important. On the other hand, in the cross-domain part, due to the lack of topic information, topic attention is dispensable. Thus, the role of token-level attention is highlighted. Overall, we can conclude that all the two key components bring certain help and benefit to the overall HNN framework.

4.3.7. Visualization of token-level attention components

As can be seen from Figure 3 and Figure 4, the performance of the framework has been improved to varying degrees after the addition of token-level attention components. To further explain the effectiveness of token-level attention components, we use visualization to analyze the function of the components in Figure 5 and Figure 6. Note that we show the stance parts (i.e. pro and con) that are unified labeled, and use a dark background to mark the boundaries. The words with an orange background denotes the con argument unit. The words with a green background denotes the pro argument unit.

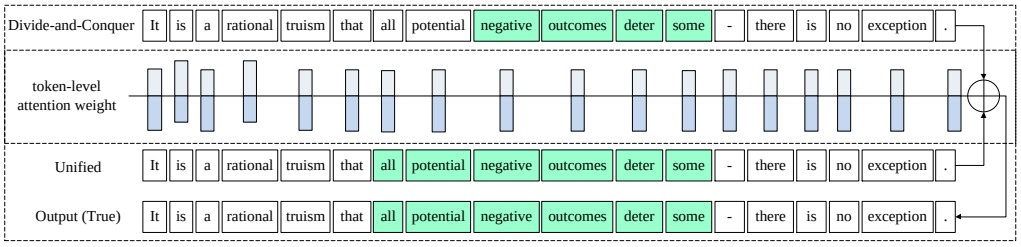

Figure 5: The visualization of attention weight for case 1.

From the Figure 5, we observe that the divide-and-conquer approach does not accurately identify the boundary of the argument unit. Nevertheless, the unified approach can make up for the shortcomings of the divide-and-conquer approach. Finally, the correct prediction results are obtained through the token-level attention component, indicating the effectiveness of the proposed framework.

Similarly, we can observe from the Figure 6 that although both the divide-and-conquer and unified approach correctly predict the boundaries of the argument unit, but the argument category of unified approach is wrong. In the process of using token-level component fusion, the divide-and-conquer approach has a larger

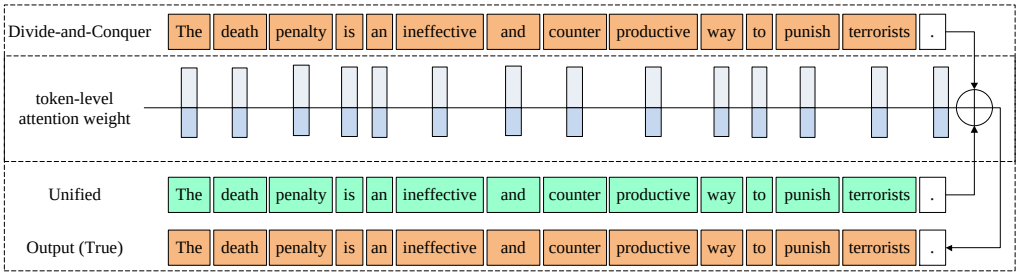

Figure 6: The visualization of attention weight for case 2.

weight, which guides the model to make correct predictions. By analyzing the visualization results, we conclude that the token-level attention component plays a critical integrating role, leading to overall better model performance.

4.3.8. Case studies

In this subsection, we select examples from test set to demonstrate the effectiveness of the proposed framework in Table 8.

Table 8: Case Studies. The "sentence" column shows the output of the AURC task from the different models, but note that we show the stance parts (i.e. PRO and CON) that are unified labeled, and use a dark background to mark the boundaries. The words with an orange background denotes the CON argument unit. The words with a green background denotes the PRO argument unit.

Model	**Sentence**	
BERT-Li-unified	However , the babies seemingly have no right to protection or life themselves because of the argument regarding when a fetus is determined be human and have life .	error
BERT-Li-unified-HNN	However , the babies seemingly have no right to protection or life themselves because of the argument regarding when a fetus is determined be human and have life.	correct
BERT-BiGRU-CRF	The death penalty is an ineffective and counter - productive way to punish terrorists .	error
BERT-BiGRU-CRF-HNN	The death penalty is an ineffective and counter - productive way to punish terrorists .	correct
BERT-IDCNN-CRF	Supporters say that few innocent people are executed and DNA testing will make convictions safer Since 1973 , over 130 people have been released from death rows throughout the country due to evidence of their wrongful convictions .	error
BERT-IDCNN-CRF-HNN	Supporters say that few innocent people are executed and DNA testing will make convictions safer Since 1973 , over 130 people have been released from death rows throughout the country due to evidence of their wrongful convictions .	correct

We observe that without the help of the HNN framework, the model may easily make wrong decisions. On the other hand, using only the unified method does not handle all cases well. Only when combining unified method with the divide-and-conquer method, the overall model performance is significantly improved. Using the HNN framework to a large extent solves these problems of inaccurate boundary recognition, wrong classification of argument units, and simultaneous occurrence of multiple different categories of argument units. Through analyzing the case study results, we can conclude that both the divide-and-conquer approach and token-level attention mechanisms play a corresponding auxiliary task roles very well under the proposed HNN framework.

5. Conclusion

AURC is a critical and practical research topic in argument mining domain which can help users identify fine-grained arguments and corresponding stances simultaneously. This paper proposes a new hierarchical neural network (HNN) framework that effectively integrates the advantages of two working methods, namely *divide-and-conquer approach* and *unified approach*. In particular, the proposed divide-and-conquer approach divides the overall AURC task into two inherent subtasks, namely, AUR and AUC, and design new methods to tackle them individually and assemble them for accurate prediction. Finally, we integrate the probability distributions obtained by the proposed divide-and-conquer approach and existing unified approach at the token-level through attention mechanism. Extensive experimental results on AURC-8 and SECA demonstrate the effectiveness of the proposed HNN framework.

Theoretically, the HNN framework can be used for other sequence labeling problems, such as NER, opinion target extraction and sentiment polarity prediction, span-level emotion cause analysis. However, due to the characteristics of the AURC task, the HNN framework cannot be perfectly adapted. For example, the AURC task contains topic information, and the argument unit is too long, etc. As such, when adapting our HNN framework to different problems, it needs to be modified properly according to their characteristics. In fact, how to extend the proposed generic framework for other NLP tasks is our future work. In addition, during the training process, argument spans for different texts need to be extracted separately, and correspondingly the time complexity is high. Therefore, how to optimize our model by taking the time complexity and training cost into consideration is also the focus of the future research.

Acknowledgements

The authors would like to thank all anonymous reviewers for their valuable comments and suggestions which have significantly improved the quality and presentation of this paper. The works described in this paper are supported by the National Natural Science Foundation of China (62076158, 62072294, 62106130, 61906112), Natural Science Foundation of Shanxi Province, China (2021030212 4084), Scientific and Technological Innovation Programs of Higher Education Institutions in Shanxi (2021L284).

References

[1] Afan, H.A., Osman, A.I.A., Essam, Y., Ahmed, A.N., Huang, Y.F., Kisi, O., Sherif, M., Sefelnasr, A., wing Chau, K., El-Shafie, A., 2021. Modeling the fluctuations of groundwater level by employing ensemble deep learning techniques. Engineering Applications of Computational Fluid Mechanics 15, 1420–1439.

[2] Ajjour, Y., Chen, W.F., Kiesel, J., Wachsmuth, H., Stein, B., 2017. Unit segmentation of argumentative texts, in: Proceedings of the 4th Workshop on Argument Mining, pp. 118–128.

[3] Banan, A., Nasiri, A., Taheri-Garavand, A., 2020. Deep learning-based appearance features extraction for automated carp species identification. Aquacultural Engineering 89, 102053.

[4] Cabrio, E., Villata, S., 2012. Natural language arguments: A combined approach, in: Proceedings of the 20th European Conference on Artificial Intelligence, Montpellier, France.

[5] Chen, C., Zhang, Q., Kashani, M.H., Jun, C., Bateni, S.M., Band, S.S., Dash, S.S., Chau, K.W., 2022. Forecast of rainfall distribution based on fixed sliding window long short-term memory. Engineering Applications of Computational Fluid Mechanics 16, 248–261.

[6] Chen, X., Hai, Z., Wang, S., Li, D., Wang, C., Luan, H., 2021. Metaphor identification: A contextual inconsistency based neural sequence labeling approach. Neurocomputing 428, 268–279.

[7] Daxenberger, J., Eger, S., Habernal, I., Stab, C., Gurevych, I., 2017. What is the essence of a claim? cross-domain claim identification, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 2055–2066.

[8] Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 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, pp. 4171–4186.

[9] Eger, S., Daxenberger, J., Gurevych, I., 2017. Neural end-to-end learning for computational argumentation mining, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 11–22.

[10] Fan, Y., Xu, K., Wu, H., Zheng, Y., Tao, B., 2020. Spatiotemporal modeling for nonlinear distributed thermal processes based on kl decomposition, mlp and lstm network. IEEE Access 8, 25111–25121.

[11] Freeman, J.B., 2011. Argument structure: Representation and theory. volume 18. Springer Science & Business Media.

[12] Fromm, M., Faerman, E., Seidl, T., 2019. Tacam: Topic and context aware argument mining, in: Proceedings of the 2019 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 99–106.

[13] Fu, Y., Li, X., Li, Y., Wang, S., Li, D., Liao, J., Zheng, J., 2022. Incorporate opinion-towards for stance detection. Knowledge-Based Systems 246, 108657.

[14] Fu, Y., Liao, J., Li, Y., Wang, S., Li, D., Li, X., 2021. Multiple perspective attention based on double bilstm for aspect and sentiment pair extract. Neurocomputing 438, 302–311.

[15] Ghazi, D., Inkpen, D., Szpakowicz, S., 2015. Detecting emotion stimuli in emotion-bearing sentences, in: Proceedings of the International Conference on Computational Linguistics and Intelligent Text Processing, pp. 152–165.

[16] Goudas, T., Louizos, C., Petasis, G., Karkaletsis, V., 2014. Argument extraction from news, blogs, and social media, in: Proceedings of the Hellenic Conference on Artificial Intelligence, Springer. pp. 287–299.

[17] Habernal, I., Gurevych, I., 2017. Argumentation mining in user-generated web discourse. Computational Linguistics 43, 125–179.

[18] Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv:1412.6980 .

[19] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C., 2016. Neural architectures for named entity recognition, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 260–270.

[20] Lauscher, A., Glavaš, G., Ponzetto, S.P., 2018. An argument-annotated corpus of scientific publications, in: Proceedings of the 5th Workshop on Argument Mining, pp. 40–46.

[21] Li, J., Fei, H., Liu, J., Wu, S., Zhang, M., Teng, C., Ji, D., Li, F., 2022. Unified named entity recognition as word-word relation classification, pp. 10965–10973.

[22] Li, M., Gao, Y., Wen, H., Du, Y., Liu, H., Wang, H., 2017. Joint rnn model for argument component boundary detection, in: Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 57–62.

[23] Li, X., Bing, L., Li, P., Lam, W., 2019. A unified model for opinion target extraction and target sentiment prediction, in: Proceedings of the Association for the Advancement of Artificial Intelligence Conference on Artificial, pp. 6714–6721.

[24] Li, X., Gao, W., Feng, S., Zhang, Y., Wang, D., 2021. Boundary detection with BERT for span-level emotion cause analysis, in: Findings of the Association for Computational Linguistics, pp. 676–682.

[25] Mochales, R., Moens, M.F., 2011. Argumentation mining. Artificial Intelligence and Law 19, 1–22.

[26] Moens, M.F., 2013. Argumentation mining: Where are we now, where do we want to be and how do we get there?, in: Post-Proceedings of the 4th and 5th Workshops of the Forum for Information Retrieval Evaluation.

[27] Moens, M.F., Boiy, E., Palau, R.M., Reed, C., 2007. Automatic detection of arguments in legal texts, in: Proceedings of the 11th International Conference on Artificial Intelligence and Law, p. 225–230.

[28] Palau, R.M., Moens, M.F., 2009. Argumentation mining: The detection, classification and structure of arguments in text, in: Proceedings of the 12th International Conference on Artificial Intelligence and Law, p. 98–107.

[29] Peldszus, A., Stede, M., 2015. Joint prediction in MST-style discourse parsing for argumentation mining, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 938–948.

[30] Petasis, G., 2019. Segmentation of argumentative texts with contextualised word representations, in: Proceedings of the 6th Workshop on Argument Mining, pp. 1–10.

[31] Sardianos, C., Katakis, I.M., Petasis, G., Karkaletsis, V., 2015. Argument extraction from news, in: Proceedings of the 2nd Workshop on Argumentation Mining, pp. 56–66.

[32] Shamshirband, S., Rabczuk, T., Chau, K.W., 2019. A survey of deep learning techniques: Application in wind and solar energy resources. IEEE Access 7, 164650–164666.

[33] Stab, C., Gurevych, I., 2014a. Annotating argument components and relations in persuasive essays, in: Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers, pp. 1501–1510.

[34] Stab, C., Gurevych, I., 2014b. Identifying argumentative discourse structures in persuasive essays, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 46–56.

[35] Stab, C., Gurevych, I., 2017. Parsing argumentation structures in persuasive essays. Computational Linguistics 43, 619–659.

[36] Stab, C., Kirschner, C., Eckle-Kohler, J., Gurevych, I., 2014. Argumentation mining in persuasive essays and scientific articles from the discourse structure perspective, in: ArgNLP.

[37] Stab, C., Miller, T., Schiller, B., Rai, P., Gurevych, I., 2018. Cross-topic argument mining from heterogeneous sources, in: Proceedings of the 2018

Conference on Empirical Methods in Natural Language Processing, pp. 3664–3674.

[38] Strubell, E., Verga, P., Belanger, D., McCallum, A., 2017. Fast and accurate entity recognition with iterated dilated convolutions, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 2670–2680.

[39] Sun, Q., Wang, Z., Zhu, Q., Zhou, G., 2018. Stance detection with hierarchical attention network, in: Proceedings of the 27th International Conference on Computational Linguistics, pp. 2399–2409.

[40] Thomas, S.N., 1981. Practical reasoning in natural language.

[41] Toulmin, S.E., 1958. The uses of argument .

[42] Trautmann, D., Daxenberger, J., Stab, C., Schütze, H., Gurevych, I., 2020. Fine-grained argument unit recognition and classification, in: Proceedings of the Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence, pp. 9048–9056.

[43] Walton, D., 2009. Argumentation Theory: A Very Short Introduction. pp. 1–22.

[44] Wang, W.c., Du, Y.j., Chau, K.w., Xu, D.m., Liu, C.j., Ma, Q., 2021. An ensemble hybrid forecasting model for annual runoff based on sample entropy, secondary decomposition, and long short-term memory neural network. Water Resources Management 35, 4695–4726.

[45] Wyner, A., Mochales-Palau, R., Moens, M.F., Milward, D., 2010. Approaches to text mining arguments from legal cases. pp. 60–79.

[46] Yan, H., Sun, Y., Li, X., Qiu, X., 2022. An embarrassingly easy but strong baseline for nested named entity recognition. arXiv preprint arXiv:2208.04534 .