Probing Structural Knowledge from Pre-trained Language Model for Argumentation Relation Classification

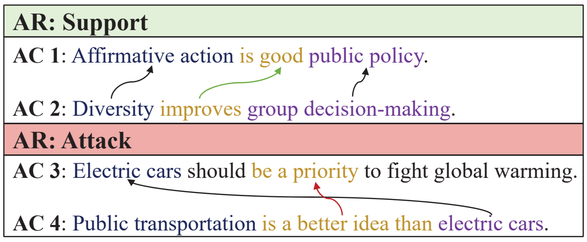
Yang Sun[1,2], **Bin Liang**[1,2], **Jianzhu Bao**[1,2], **Min Yang**[3*], **and Ruifeng Xu**[1,2,4*]
[1] Harbin Institute of Technology, Shenzhen, China
[2] Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies
[3] SIAT, Chinese Academy of Sciences, Shenzhen, China
[4] Peng Cheng Laboratory, Shenzhen, China
sy95@mail.ustc.edu.cn, bin.liang@stu.hit.edu.cn
jianzhubao@gmail.com, min.yang@siat.ac.cn, xuruifeng@hit.edu.cn

Abstract

Extracting fine-grained structural information between argumentation component (AC) pairs is essential for argumentation relation classification (ARC). However, most previous studies attempt to model the relationship between AC pairs using AC level similarity or semantically relevant features. They ignore the complex interaction between AC pairs and cannot effectively reason the argumentation relation deeply. Therefore, in this paper, we propose a novel dual prior graph neural network (DPGNN) to jointly explore the probing knowledge derived from pre-trained language models (PLMs) and the syntactical information for comprehensively modeling the relationship between AC pairs. Specifically, we construct a probing graph by using probing knowledge derived from PLMs to recognize and align the relational information within and across the argumentation components. In addition, we propose a mutual dependency graph for the AC pair to reason the fine-grained syntactic structural information, in which the syntactical correlation between words is set by the dependency information within AC and the mutual attention mechanism across ACs. The knowledge learned from the probing graph and the dependency graph are combined to comprehensively capture the aligned relationships of AC pairs for improving the results of ARC. Experimental results on three public datasets show that DPGNN outperforms the state-of-the-art baselines by a noticeable margin.

1 Introduction

Argumentation relation classification (ARC) is the most challenging subtask of argumentation mining, which requires the model to understand complex linguistic interactions between argumentation components (Lawrence and Reed, 2020). The goal of ARC is to identify the argumentation relation (i.e., SUPPORT or ATTACK) between argumentation components (AC) pairs. As shown in Figure 1, there is an example with support relation and an example with attack relation, where AC2 supports AC1 and AC4 attacks AC3.


AR: Support

AC 1: Affirmative action is good public policy.

AC 2: Diversity improves group decision-making.

AR: Attack

AC 3: Electric cars should be a priority to fight global warming.

AC 4: Public transportation is a better idea than electric cars.

Figure 1: Examples of argumentation relation over AC pairs , where the words with different colors represent the structure within AC and the solid directed lines denote the fine-grained alignment of structure within AC pair.

ARC needs the model to effectively explore structural knowledge within and cross ACs so as to better infer the argumentation relation between AC pairs. Intuitively, the semantic structure can capture the correlations among semantically similar words, while the syntactic structure can capture the syntactical constraint (e.g., dependency information) for syntactically relevant words. Taking Figure 1 as an example, the words "affirmative action" and "is good" in AC1 can be aligned with "diversity" and "improves" in AC2 respectively for identifying the support relation. However, most previous efforts (Palau and Moens, 2009; Peldszus, 2014; Cocarascu and Toni, 2017; Galassi et al., 2018) merely focus on the AC-level similarity between AC pairs and result in sub-optimal performance. Although recent works (Gemechu and Reed, 2019; Jo et al., 2021) model the fine-grained semantically relevant features (such as words) between AC pair by introducing external knowledge, they ignore the complex interactions of AC pair and cannot effectively reason the argumentation relation deeply. For instance, in Figure 1, previous works only empha-

[*] Min Yang and Ruifeng Xu are corresponding authors

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 3605–3615
December 7-11, 2022 ©2022 Association for Computational Linguistics

size the superficial similarity between individual words "better" and "priority" and fail to capture the overall opposition relation between AC3 and AC4 since "electric cars" is the object in AC3 but the subject in AC4. To our best knowledge, the structural knowledge within or across ACs has not been simultaneously investigated for the Argumentation Relation Classification.

In this paper, we propose a dual prior graph neural network (DPGNN) to leverage two prior knowledge, i.e., dependency information and probing knowledge from pre-trained language models (PLMs), by constructing dual graph modules for ARC. By combining the information from the two graph modules, DPGNN can more accurately capture the fine-grained relations between AC pairs. Specifically, we construct the mutual dependency graph from two perspectives (i.e., intra-AC perspective and inter-AC perspective) to gain the syntactical structure information with dependency parsing and attention mechanisms. The intra-AC graph constructs the syntactical structure within ACs in the mutual graph, and the inter-AC graph aligns and builds the structure between AC pair to explore the argumentation relation. Despite the effectiveness of the learned dependency information, it is difficult to recognize semantically relevant words such as synonyms and antonyms by only relying on the dependency information.

To complement the dependency information for ARC, we probe the relational knowledge from PLMs, such as BERT (Kenton and Toutanova, 2019) and RoBERTa (Liu et al., 2020), which contain rich semantic knowledge for ARC, such as the counterpart relationship between "public policy" and "group decision-making" as shown in Figure 1. Concretely, we first probe token representations and attention matrices from PLMs, where the token representations form the nodes of the probing graph and the attention matrices form the edges between nodes. In particular, to obtain useful probing knowledge with respect to ARC, we propose three different levels of probes[1] (i.e., word-, AC- and pair-level) with adaptive attention mechanisms for probing token representations and attention matrices. The key idea behind this probing technique is motivated by the observation that the knowledge stored in PLMs contains rich linguistic and relational knowledge (Petroni et al., 2019; Zhong et al., 2021) about words (e.g., synonyms and antonyms) in PLMs (Jawahar et al., 2019; Clark et al., 2019). We disentangle each probed attention matrix into four attention sub-matrices according to the span of each AC within AC pair to construct the probing graph from intra-AC and inter-AC perspectives, respectively. Finally, we combine the graph representations from the dependency graph and the probing graph with a biaffine function (Morio et al., 2020) for argumentation relation prediction.

Our main contributions are three-fold. (1) We propose a dual prior graph neural network for ARC, which jointly explores the probed knowledge derived from PLMs and the syntactical information to comprehensively model the relationship between AC pairs. (2) We probe rich relation knowledge from PLMs in terms of AC-pair level, which elicit relations to robustly capture the semantic correspondences between ACs and AC pairs. (3) We conduct extensive experiments on three benchmark ARC datasets. Experimental results show that our method significantly outperforms previous methods and achieves new state-of-the-art results.

2 Related Work

2.1 Argumentation Relation Classification

Early works (Palau and Moens, 2009; Wyner et al., 2010; Cabrio and Villata, 2012; Peldszus, 2014; Peldszus and Stede, 2015) focused on argumentative relation classification based on several discrete features involving grammar and text statistics for ARC in specific corpus. Previous studies have mostly employed traditional methods such as support vector machines, naive Bayes classifiers and maximum entropy classifiers. With the widespread usage of deep learning, Cocarascu and Toni (2017) proposed a deep learning architecture based on Long-Short Term Memory (LSTM) networks for ARC. Galassi et al. (2018) explored residual networks for ARC. Previous research works only focused on the content of argumentation component pairs to determine relationships between AC pairs. To capture more precise semantic similarity between AC pair, recent works develop external knowledge and designed feature for ARC. For example, Gemechu and Reed (2019) designed four fine-grained features and identifies argumentation relations by exploiting the similarity among the four fine-grained features. Paul et al. (2020) extracted the relevant knowledge from the general knowledge resource ConceptNet and encoded ACs

[1] A probe is a simple neural network that develops the features (i.e. hidden states and attention weights) from PLM for specific task (Wu et al., 2020)

and knowledge using two BiLSTM with attention mechanism for ARC. Jo et al. (2020) used BERT with four characteristics regarding the sentence's content, proposition types, tone, and an external engineering knowledge source for detecting attackable sentences. Jo et al. (2021) classified argumentative relation based on multiple designed features including factual consistency, sentiment coherence, causal relation and normative relation between two ACs. In this paper, we emphasize the ARC task based on the form of AC pairs. Different from previous works, we do not explicitly introduce external knowledge, but integrate the probed knowledge elicited from PLMs to learn fine-grained reasoning for ARC.

2.2 Probing Knowledge from PLMs

The success of PLMs has led to a large number of studies eliciting the rich knowledge that PLMs learn implicitly during pre-training (Jawahar et al., 2019; Clark et al., 2019; Wu et al., 2020). Some works probe PLMs with a small amount of learnable parameters considering a variety of linguistic properties, such as morphology (Belinkov et al., 2017), word sense (Reif et al., 2019), syntax (Hewitt and Manning, 2019; Dai et al., 2021). There are also some works (Petroni et al., 2019; Zhong et al., 2021) that seek to answer to what extent the PLMs store factual, relational and commonsense knowledge. Wang et al. (2022) elicited relational structures from PLMs via a probing procedure and utilized the induced relations to augment the graph-based text-to-SQL parsers for better schema linking. Different from previous studies, we aim to probe rich relational knowledge for ARC.

2.3 Graph Neural Network for NLP

The recent success of graph neural networks (GNN) has boosted research in natural language processing (NLP) tasks, such as fact verification (Zhong et al., 2020) and aspect-based sentiment analysis (Liang et al., 2022). Recently, HARGAN (Huang et al., 2021) introduces the argumentation relation information for persuasiveness prediction with a GNN-based model. The previous works do not fully explore the fine-grained graph structure in ARC. In this paper, we propose a dual prior GNN to align the fine-grained structure information between AC pair by introducing probing knowledge from PLM and dependency information.

3 Methodology

Task Definition Following previous work (Jo et al., 2021), we assume an AC pair (P, Q) are given in an argumentative text, where the argumentation components $P = (p_1, p_2, \dots, p_m)$ and $Q = (q_1, q_2, \dots, q_n)$ consist of m and n tokens, respectively. The goal of ARC is to predict the relation type $y_{(P,Q)}$ (i.e., *Support* or *Attack*).

Model Overview Figure 2 illustrates the architecture of our DPGNN model. Our method leverages two complementary structural knowledge to construct dual graphs for learning the alignment of fine-grained structures within ACs and between AC pairs for ARC. Concretely, we propose three probes to elicit knowledge from hidden states and attention matrices in BERT. The probed hidden states and attention matrices are employed to construct a probing graph for reasoning the relation between the AC pair. In addition, we develop the dependency graph to gain the syntactical structure information with dependency parsing and attention mechanisms. Finally, a biaffine module is devised to combine the probing graph and the mutual dependency graph for improving the performance of ARC.

3.1 Probing Knowledge from PLMs

3.1.1 Probing Knowledge

Pre-trained language models (PLMs) contain a rich hierarchy of linguistic information in internal vector representations (i.e., hidden states) (Jawahar et al., 2019). The self-attention layers in BERT contain not only long-distance dependencies between words within each AC but also rich relational knowledge (Clark et al., 2019) for reasoning. In this paper, we aim to probe hidden states and attention matrices in PLM (i.e., BEER) to capture the alignment of structures between each AC pair. Specifically, we develop three probes from the word-level, AC-level, and AC pair-level respectively, where each word, AC, and AC pair represent the probing units (smallest units) for the three corresponding probes. The word-, AC- and AC pair-level probes elicit the hidden states and attention matrices from PLMs.

The input of BERT is the AC pair (P, Q), which is formulated as "[CLS]P[SEP]Q[SEP]". We define the hidden states of the AC pair (P, Q) in the BERT layers (i.e., 12 layers in the BERT-base version) as $B = \{H_1^B, \dots, H_{12}^B\}$, where

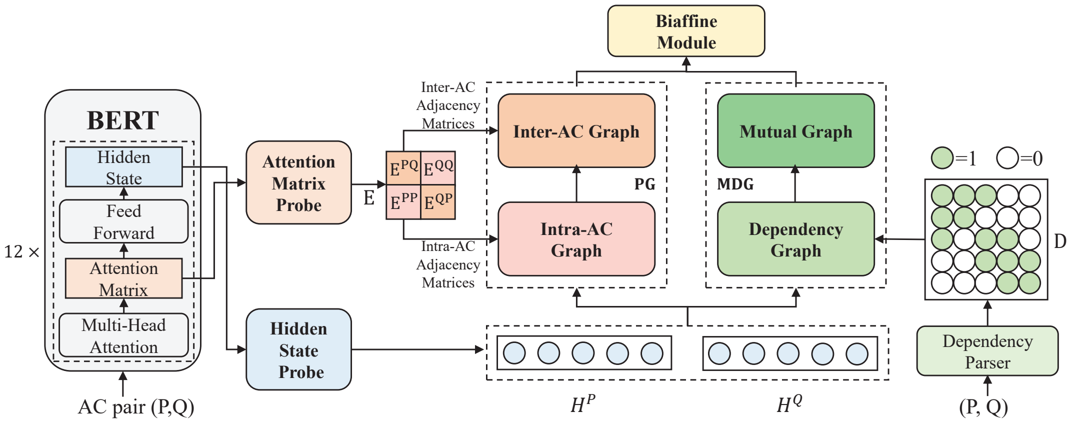


Figure 2: The architecture of DPGNN, where "PG" and "MDG" represent probing graph and mutual dependency graph, respectively.

$H_i^B = \{\mathbf{h}_{i,1}^B, \ldots, \mathbf{h}_{i,m+n}^B\}$ is a concatenation of the hidden states $H_i^P = \{\mathbf{h}_{i,1}^P, \ldots, \mathbf{h}_{i,m}^P\}$ and $H_i^Q = \{\mathbf{h}_{i,1}^Q, \ldots, \mathbf{h}_{i,n}^Q\}$ of P and Q respectively. In a similar way, we denote the attention matrices in the BERT layers (i.e., 12 layers) as $A = \{E_1^A, \ldots, E_{12}^A\}$ with the input of the AC pair (P, Q), where $E_i^A = \{\mathbf{e}_{i,1}^A, \ldots, \mathbf{e}_{i,m+n}^A\}$ is the average value of the original attention matrix along with the head dimension and $\mathbf{e}_{i,j}^A \in \mathbb{R}^{m+n}$. Here, $E_i^A = [E_i^P; E_i^Q]$ where $E_i^P = \{\mathbf{e}_{i,1}^P, \ldots, \mathbf{e}_{i,m}^P\}$ and $E_i^Q = \{\mathbf{e}_{i,1}^Q, \ldots, \mathbf{e}_{i,n}^Q\}$ represent the attention matrices of P and Q respectively.

AC Pair-Level Probe We propose an AC pair-level probe to capture the interaction and alignment between a pair of ACs by using the AC pair as the probing unit. In particular, we first calculate the representation for the AC pair as $\mathbf{h}_i^{\mathrm{APL}} = \frac{1}{m}\sum_{j=1}^{m+n} \mathbf{h}_{i,j}^B$ in the i-th layer by applying the average pooling over H_i^B. In addition, we also calculate the attention vector $\mathbf{e}_i^{\mathrm{APL}}$ for the AC pair (P,Q) in the i-th layer via average pooling: $\mathbf{e}_i^{\mathrm{APL}} = \frac{1}{m}\sum_{j=1}^{m} \mathbf{e}_{i,j}^A$.

Probing Knowledge across Layers To adaptively choose the important features across layers, we apply a soft attention mechanism with two layer feedback neural network (Wang et al., 2019) to learn the combined representations and attention matrices. Formally, we first calculate the learned weight α_i^{APL} of representation $\mathbf{h}_i^{\mathrm{APL}}$ of AC pair (P, Q) in the i-th layer as follows:

$$\alpha_i^{\mathrm{APL}} = \mathbf{a}_\alpha \cdot \tanh(\mathbf{W}_\alpha \mathbf{h}_i^{\mathrm{APL}} + \mathbf{b}_\alpha) \qquad (1)$$

where $\mathbf{W}_\alpha$, $\mathbf{b}_\alpha$ and $\mathbf{a}_\alpha$ are learnable parameters. After that, we normalize the attention weights via softmax function and get the normalized coefficient $\tilde{\alpha}_i^{\mathrm{APL}}$ that is easily comparable across different layers. Although the coefficient $\tilde{\alpha}_i^{\mathrm{APL}}$ is formulated by the AC pair representation, our goal is to gain the fine-grained word representations. Thus, all words in AC pair (P, Q) share the same coefficient $\tilde{\alpha}_i^{\mathrm{APL}}$ in the i-th layer. With the learned weights as coefficients, we fuse the representations of j-th word in all layers to obtain the probing representation as follows:

$$\mathbf{h}_j = \sum_{i=1}^{12} \tilde{\alpha}_i^{\mathrm{APL}} \mathbf{h}_{i,j}^B \qquad (2)$$

Finally, we obtain the probing representations $H^P = \{\mathbf{h}_1^P, \ldots, \mathbf{h}_m^P\}$ and $H^Q = \{\mathbf{h}_1^Q, \ldots, \mathbf{h}_n^Q\}$ of AC pair (P, Q). The similar equations 1-2 is also applied in probing attention matrices, and we obtain the probing attention vectors of all words in AC pair to form the probing attention matrix $E = \{\mathbf{e}_1, \ldots, \mathbf{e}_{m+n}\}$, where $\mathbf{e}_i \in \mathbb{R}^{m+n}$.

3.1.2 Probing Graph Construction

We construct the probing graph using probing knowledge including representations and an attention matrix. The probing graph takes the unique words in the AC pair as vertices and the embeddings of the vertices are initialized with the probing representations. To effectively construct and align the relational structure within ACs and between AC pairs, we propose intra-AC and inter-AC graphs by using the probing attention matrix to build the weighted edges within ACs and between AC pairs in the probing graph separately.

There are four attention sub-matrices in the probing attention matrix E of AC pair (P, Q) as shown in the left part of Figure 2, which represent the word correlation within AC and between AC respectively. Thus, we separate the probing attention matrix E into four attention sub-matrices $E^{PP} \in \mathbb{R}^{m \times m}$, $E^{QQ} \in \mathbb{R}^{n \times n}$, $E^{PQ} \in \mathbb{R}^{m \times n}$ and $E^{QP} \in \mathbb{R}^{n \times m}$ of AC pair (P, Q) to build and align the relational structure within ACs and between ACs, where E^{PP} and E^{QQ} are denoted by intra-AC adjacency matrices, and E^{PQ} and E^{QP} are denoted by inter-AC adjacency matrices.

Intra-AC Graph Construction Intuitively, the relational structure within AC can build the correlation among semantically similar words within AC and help the alignment between AC pair for ARC. Thus, we first design an intra-AC graph to build the relational structure based on the intra-AC adjacency matrix for each AC in the probing graph. The intra-AC adjacency matrix has captured semantically related terms of each word in AC. Specifically, given the AC P, we built an intra-AC graph $\mathcal{G}_{\mathrm{P}}^{\mathrm{pro}}$ with the node representations H^P. The intra-AC adjacency matrix E^{PP} is set to the initial weighted edges to form the relational structure within the AC. Then we normalize the intra-AC adjacency matrix E^{PP} of AC P via softmax function as the normalized adjacency matrix $\tilde{E}^{PP} = \mathrm{softmax}(E^{PP})$ within AC P for easily comparable across different words because the intra-AC adjacency matrices are segmented from an attention matrix: Then the updated node representations $S^P = \{\mathbf{s}_i^P, \dots, \mathbf{s}_m^P\}$ can be obtained by the following equation:

$$\mathbf{s}_i^P = \tilde{\mathbf{E}}_i^{PP} H^P \tag{3}$$

where $\tilde{\mathbf{E}}_i^{PP}$ is the i-th row in $\tilde{E}^{PP}$. In this way, the structure information within AC could be extracted into the node representations via the intra-AC graph. The same equations can be applied for AC Q to acquire the updated node representations $S^Q = \{\mathbf{s}_i^Q, \dots, \mathbf{s}_m^Q\}$.

Inter-AC Graph Construction To explore the complex interaction and relationship between the AC pair, we utilize inter-AC adjacency matrices with prior relational knowledge to build an intra-AC graph to align the fine-grained node representations between AC pairs. Formally, given the intra-AC graph $\mathcal{G}_{\mathrm{P}}^{\mathrm{pro}}$ as query and the intra-AC graph $\mathcal{G}_{\mathrm{Q}}^{\mathrm{pro}}$ as value, we produce the normalized adjacency matrix $\tilde{E}^{PQ} = \mathrm{softmax}(E^{PQ})$ for the query and value pair. After that, we calculate $\mathcal{G}_{\mathrm{P}}^{\mathrm{pro}}$-specific node representations $C^P = \{\mathbf{c}_1^P, \dots, \mathbf{c}_m^P\}$ are formulated as:

$$\mathbf{c}_i^P = \tilde{\mathbf{E}}_i^{PQ} S^Q \tag{4}$$

where $\tilde{\mathbf{E}}_i^{PQ}$ is the i-th row in $\tilde{E}^{PQ}$. We apply alignment function (Shen et al., 2018) to perform fine-grained node-to-node alignment and calculate aligned node representations $V^P = \{\mathbf{v}_1^P, \dots, \mathbf{v}_m^P\}$ of intra-AC graph $\mathcal{G}_{\mathrm{P}}^{\mathrm{pro}}$, where $\mathbf{v}_i^P$ is calculated as:

$$\mathbf{v}_i^P = \mathbf{W}_{\mathbf{v}}[\mathbf{s}_i^P, \mathbf{c}_i^P, \mathbf{s}_i^P - \mathbf{c}_i^P, \mathbf{s}_i^P \odot \mathbf{c}_i^P] \tag{5}$$

where $\mathbf{W}_{\mathbf{v}}$ is a weight matrix and $\odot$ denotes element-wise multiplication.

Similarly, we take the intra-AC graph $\mathcal{G}_{\mathrm{Q}}^{\mathrm{pro}}$ as query and the intra-AC graph $\mathcal{G}_{\mathrm{P}}^{\mathrm{pro}}$ as value. By using Equations 4-5, we can calculate the aligned node representations $V^Q = \{\mathbf{v}_1^Q, \dots, \mathbf{v}_n^Q\}$ of the intra-AC graph $\mathcal{G}_{\mathrm{Q}}^{\mathrm{pro}}$. Then, we apply mean pooling for the aligned vector V^P and V^Q to calculate the relation-specific intra-AC graph representations $\mathbf{g}_P^{\mathrm{pro}}$ for $\mathcal{G}_{\mathrm{P}}^{\mathrm{pro}}$ and $\mathbf{g}_Q^{\mathrm{pro}}$ for $\mathcal{G}_{\mathrm{Q}}^{\mathrm{pro}}$ by:

$$\mathbf{g}_P^{\mathrm{pro}} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{v}_i^P, \quad \mathbf{g}_Q^{\mathrm{pro}} = \frac{1}{n}\sum_{j=1}^{n} \mathbf{v}_j^Q \tag{6}$$

3.2 Mutual Dependency Graph Construction

The syntax is the grammatical structure of the text (e.g., dependency tree), whereas semantics represent the meaning being conveyed. Thus, the syntactic structure can help the model capture the long-term and syntactically relevant contextual words as clues to reason argumentation relations, which are difficult to be learned by semantic-based methods. To construct the syntactic structure within sentences, Liang et al. (2021a) propose a dependency graph neural network, but they cannot capture the complex interaction between AC pairs. Thus, we propose a mutual dependency graph from the intra-AC perspective (i.e., dependency graph) and inter-AC perspective (i.e., mutual graph) which aims to build and align the syntactic structure within AC and between AC pairs by using syntactic dependency information and attention mechanism. We also construct the mutual dependency graph with unique words in AC pair as vertices initialized with the probing representations.

3.2.1 Dependency Graph Construction

The goal of the dependency graph in the mutual dependency graph is to build a syntactic structure within AC by developing the syntactical dependency. Compared with the intra-AC graph in the probing graph, the dependency graph only emphasizes the crucial syntactic word relations and evades the inconsequential ones in each AC, which can be

Dataset	Train	Dev	Test
DN	11098	472	707
DC	6581	496	330
PE	2697	362	773

Table 1: The statistics of the evaluated datasets

a hard bound of structure within ACs to mutually compensate the error word dependencies with the probing graph.

To build the syntactic structure within each AC, we construct dependency graph $\mathcal{G}^{\text{dep}}$ over the dependency tree for each AC. Given a dependency graph $\mathcal{G}_{\text{P}}^{\text{dep}}$ with the node representations H^P of AC P, the discrete adjacency matrix $D \in \mathbb{R}^{m \times m}$ for $\mathcal{G}_{\text{P}}^{\text{dep}}$ is a binary matrix and can be derived from the dependency tree [2] where $D_{i,j} = 1$ represents that i-th word is connected to j-th word in the dependency tree of the AC.

After that, we feed the node representations and the adjacency matrix D into the graph attention network (GAT) (Veličković et al., 2018) to update the representation of each word node and erect the intrinsic structure by aggregating information from its neighbors in $\mathcal{G}_{\text{P}}^{\text{dep}}$ and $\mathcal{G}_{\text{Q}}^{\text{dep}}$. Then we achieve the updated node representations $Z^P = \{\mathbf{z}_1^P, \ldots, \mathbf{z}_m^P\}$ and $Z^Q = \{\mathbf{z}_1^Q, \ldots, \mathbf{z}_n^Q\}$ with syntactic structure for the nodes within AC P and Q.

3.2.2 Mutual Graph Construction

We try to directly apply the inter-AC graph in the probing graph for aligning the syntactic structure between AC pairs but find a decrease in model performance. We suspect the reason is that an inter-AC perspective module is hard to capture heterogeneous relational information between AC pairs. To effectively coordinate the syntactic structure information within ACs, we construct a mutual graph to align the AC pairs at a fine-grained level. To the end, we employ a mutual attention mechanism $\text{att}_\beta(\cdot)$ with dot product to learn the importance score $\beta_{i,j}$ for each node i in $\mathcal{G}_{\text{P}}^{\text{dep}}$ from each node j in $\mathcal{G}_{\text{Q}}^{\text{dep}}$ as follows:

$$\beta_{i,j} = \text{att}_\beta(\mathbf{W}_P \mathbf{z}_i^P, \mathbf{W}_Q \mathbf{z}_j^Q) \tag{7}$$

where $\mathbf{W}_P$ and $\mathbf{W}_Q$ are weight matrices. Then, we normalize $\beta_{i,j}$ across all nodes in $\mathcal{G}_{\text{Q}}^{\text{dep}}$ using the softmax function to get $\tilde{\beta}_{i,j}$. Next, the $\mathcal{G}_{\text{P}}^{\text{dep}}$-specific node representations $U^P = \{\mathbf{u}_1^P, \ldots, \mathbf{u}_m^P\}$ is obtained by using the weighted sum over Z^Q, where $\mathbf{u}_i^P$ is computed by $\mathbf{u}_i^P = \sum_{j=1}^n \tilde{\beta}_{i,j} \mathbf{z}_j^Q$. Finally, the alignment function and mean pooling operation in Equations 5-6 are applied to calculate the relation-specific dependency graph representations $\mathbf{g}_P^{\text{dep}}$ for $\mathcal{G}_{\text{P}}^{\text{dep}}$ in here. The similar procedures is processed to obtain the relation-specific dependency graph representations $\mathbf{g}_Q^{\text{dep}}$ for $\mathcal{G}_{\text{Q}}^{\text{dep}}$ from $\mathcal{G}_{\text{P}}^{\text{dep}}$.

3.3 Biaffine Module

To harmonize the information from dual graphs, we use the concatenation operation on the relation-specific graph representations of the probing graph and mutual dependency graph to get the comprehensive relation-specific graph representations r^P and r^Q of AC P and Q:

$$\mathbf{r}^P = \mathbf{g}_P^{\text{pro}} || \mathbf{g}_P^{\text{dep}}, \quad \mathbf{r}^Q = \mathbf{g}_Q^{\text{pro}} || \mathbf{g}_Q^{\text{dep}} \tag{8}$$

We then apply a biaffine operation (Morio et al., 2020) to capture the bidirectional property of AC pair and a softmax function to produce the relation label probability $p(y)$:

$$p(y_{(P,Q)} | \mathbf{r}^P, \mathbf{r}^Q) = \text{softmax}\Big(\varrho(\mathbf{r}^P, \mathbf{r}^Q)\Big) \tag{9}$$

where $y_{(P,Q)}$ is the ground-truth relation label of AC pair, $\varrho(\mathbf{x}, \mathbf{y}) = [\begin{smallmatrix} \mathbf{x} \\ 1 \end{smallmatrix}]^\top \mathbf{W}_\varrho \mathbf{y}$ and $\mathbf{W}_\varrho$ is learnable weights.

3.4 Loss Function

Our training goal is to minimize the following total objective function:

$$\mathcal{L} = -\sum_D \ln p(y_{(P,Q)}) + \lambda ||\theta||_2 \tag{10}$$

where D denotes the training dataset, θ represents all trainable parameters, and λ is the coefficient of the regularization term.

4 Experimental Setup

4.1 Datasets

In order to evaluate the performance of our DPGNN model, we conduct experiments on three public benchmark datasets including debatepedia-normative (DN), debatepedia-casual (DC) (Jo et al., 2021) and PE (Stab and Gurevych, 2017) and follow their official train/dev/test split. The detailed statistics of three datasets are shown in Table 1.

[2] In this work, we use spaCy toolkit for generating dependency tree of the input sentence: https://spacy.io/.

Model	DN				DC			
	ACC	**Macro**	**Support-F1**	**Attack-F1**	**ACC**	**Macro**	**Support-F1**	**Attack-F1**
BiLSTM	71.0	71.0	71.3	70.7	68.5	68.3	71.0	65.5
LSTM-ATT	71.6	71.5	70.1	72.9	70.3	70.3	71.2	69.4
Hybrid Net	67.2	67.2	68.1	66.3	59.7	58.8	64.5	53.2
BERT	79.1	79.4	79.8	79.0	80.7	80.7	81.4	79.9
BERT+LX	78.4	78.4	79.2	77.5	81.6	81.5	82.3	80.8
BERT+MT	79.6	79.6	80.0	79.1	77.6	77.5	77.5	78.9
LogBERT	81.0	80.7	81.1	80.4	81.2	80.8	81.7	80.0
DPGNN	**82.9**	**82.9**	**82.3**	**83.5**	**84.2**	**84.1**	**85.6**	**82.6**

Table 2: Performance comparison on DN and DC datasets. Our improvements over baselines are statistically significant with $p < 0.05$.

Model	ACC	Macro	Support-F1	Attack-F1
BiLSTM	93.8	55.5	96.8	14.2
LSTM-ATT	91.7	55.7	95.6	15.8
Hybrid Net	92.9	55.8	96.3	15.4
BERT	93.3	60.0	96.5	23.5
DPGNN	**94.5**	**63.8**	96.6	**31.0**

Table 3: Performance comparison on PE dataset

4.2 Evaluation Metrics

We apply the same evaluation metrics with previous works (Bao et al., 2021; Jo et al., 2021; Liang et al., 2021b), including accuracy (ACC), per-class F_1 (denoted as Support-F1 and Attack-F1), and macro averaged score (denoted as Macro). Concretely, the macro averaged score is calculated by averaging all the per-class F_1 scores.

4.3 Baselines

We compare our model with state-of-the-art baselines:

- **BiLSTM** (Cocarascu and Toni, 2017): This model optimizes ARC using two BiLSTMs to encode AC pair, respectively.
- **LSTM-ATT** (Ma et al., 2017): It employs two LSTMs and an interaction attention to generate representations for AC pairs.
- **Hybrid-Net** (Chen et al., 2018): It encodes the input using BiLSTM and uses self- and cross-attention between words for ARC.
- **BERT** (Kenton and Toutanova, 2019): This model uses vanilla BERT model by feeding the AC pair and using the representation of [CLS] for predictions.
- **BERT+LX** (Jo et al., 2021): This model employs BERT as encoder and latent cross to incorporate external features, such as factual consistency and sentiment coherence, for ARC.
- **BERT+MT** (Jo et al., 2021): It uses multi-task learning to train the ARC and other logic tasks, such as textual entailment and sentiment classification, simultaneously.
- **LogBERT** (Jo et al., 2021): This model applies BERT as encoder to pre-train in logic tasks and fine-tune on the target dataset for ARC finally.

For the PE dataset, we do not compare our model with BERT-LX, BERT-MT, and LogBERT since they require a large number of external engineering features and annotations that cannot be easily acquired.

4.4 Implementation Details

We use PyTorch to implement the proposed model on an NVIDIA GeForce RTX 3080 GPU. We use the uncased BERT base model[3] as our PLM. Our model is optimized using AdaW (Loshchilov and Hutter, 2018) with the learning rates of 1e-5 on the BERT layers and 1e-3 on other layers on all datasets. We set the size of word embedding as 768. The default setup in probing representation and attention matrix is a pair-level probe in all datasets because of their excellent performance. For all datasets, we set the batch size as 32 and the weight decay λ as 1e-3. We adopt dropout with a dropout rate of 0.1 to avoid overfitting. The training process stops if the accuracy score does not increase for 5 epochs on the validation data. The code and data are available [4].

[3] We implement BERT using huggingface toolkit: https://huggingface.co/

[4] https://github.com/HITSZ-HLT/DPGNN

Model	DN		DC	
	Macro	∇	Macro	∇
DPGNN	**82.9**	-	**84.2**	-
-w/o MDG	81.6	-1.3	83.0	-1.2
-w/o DI	82.3	-0.6	82.1	-2.1
-w/o PG	81.9	-1.0	82.4	-1.8
-w/o PR	82.4	-0.5	83.6	-0.6

Table 4: The ablation results in terms of removing different components in DPGNN on the DN and DC datasets

5 Experimental Results

5.1 Performance Comparison

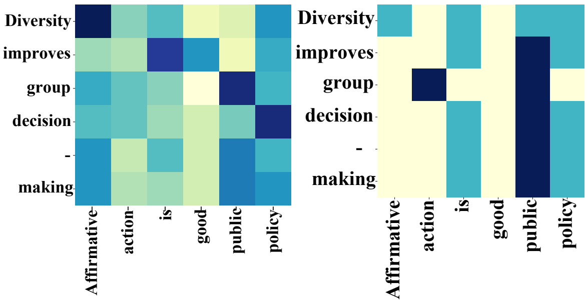
We report the results of DPGNN and compared baselines in Table 2. We can observe that our DPGNN model achieves the best performance on all the datasets. On the DC and DN datasets, our model outperforms the best performing baseline by 2.6 and 2.6, and 1.9 and 2.2 on the ARC task respectively in terms of accuracy, per-class F1 score, and macro averaged score. In addition, we also observe that LSTM-based baselines (i.e., BiLSTM, LSTM-ATT, and Hybrid Net) generally perform worse than BERT-based models. This may be because the pre-trained BERT contains rich knowledge learned from the large-scale corpora. The BERT-based methods that integrate multiple external features (i.e., BERT+LX and LogBERT) achieve slightly better performance than the original BERT model. Furthermore, DPGNN performs better than all the BERT-based models by leveraging the probing knowledge from PLMs and dependency knowledge to effectively capture the relational features between AC pairs. We observe similar trends on the PE dataset in Table 3.

5.2 Ablation Study

Effectiveness of Different Components To investigate the effectiveness of different components in DPGNN, we conduct an ablation study in terms of removing the mutual dependency graph (w/o MDG), removing dependency information (w/o DI), removing the probing graph (w/o PG), and removing probing representation (w/o PR) respectively. It is noteworthy that removing the probing graph is equal to removing the probing attention matrix. As shown in Table 4, the full model of DPGNN has the best performance. We also observe that w/o MDG and w/o PG achieve similar performance and perform worse than DPGNN, verifying the effectiveness of the probing knowledge and the dependency knowledge that can complement each other for capturing the fine-grained structural information within ACs and between AC pairs. The performance of DPGNN w/o DI slightly decreases on the DN dataset, while there is a huge performance drop on the DC dataset. This may be because the DN dataset relies on the relational structure while the syntactic structure contributes more to DC dataset including longer and more syntactically sensitive samples.

Probe	DN		DC	
	ACC	Macro	ACC	Macro
Word-Level	81.6	81.6	82.7	82.6
AC-Level	82.2	82.1	83.6	83.6
AC Pair-Level	**82.9**	**82.9**	**84.2**	**84.1**

Table 5: The ablation results in terms of applying different probes in DPGNN on the DN and DC datasets



(a) intra-AC Graph (b) Mutual Graph

Figure 3: The attention maps of an example in Figure 1 when reasoning the argumentation relation between AC1 and AC2.

Effectiveness of Different Probes We also explore two additional probes, including the word-level probe and AC-level probe. The probing unit for the word-level probe is individual words. Thus, we treat each learned hidden state and attention vector as word-level probing knowledge. The AC-level probe leverages the global information of each AC to elicit the AC-level knowledge by applying the average operation over the word representations to obtain the AC representation in each layer. Similar to AC pair-level probe, we can acquire the probing representations and attention matrices of the AC pair in the word- and AC-level probes. We report the results of our DPGNN with the three different probes. As shown in Table 5, the word-level probe always performs worse than the AC-level and AC pair-level probes on both DC and DN datasets. This may be because the word-level probe does not capture the global information of each AC and each AC pair. By contrast, the pair-level probe achieves the best performance among the three probes on the two datasets, since the pair-level probe can better capture the established association of words over AC pair, rather than merely learn the knowledge within each AC.

5.3 Case Study

We use a case study to visualize the attention maps of the inter-AC probing graph and the mutual dependency graph when predicting the relation from AC2 to AC1. The attention maps are shown in Figure 1. The color depth indicates the importance degree of the word. As shown in Figure 3a, the important words such as “affirmative” and “good” in AC1 are aligned with the words “diversity” and “improve” in AC2. Meanwhile, we can also observe that the attention weights capture the important alignment between “public policy” and “decision making”. This verifies that the inter-AC graph and mutual graph can align the rich structure between AC pairs. By combining heterogeneous information from dual graphs, our model can obtain comprehensive complementary information for effective ARC.

6 Conclusion

In this paper, we proposed a graph-based model DPGNN with two prior knowledge, i.e., probing knowledge elicited from PLM and syntactical dependency information, to model the relational and syntactic structures within ACs and between AC pairs for ARC. To effectively capture the useful probing knowledge from BERT, we propose three probes to elicit word-, AC- and pair-level knowledge. In addition, DPGNN integrated the probing graph with decoupled probing attention matrices and the mutual dependency graph with syntactic dependency information to make our model more effective to utilize the heterogeneous structure within ACs and between AC pairs. Experimental results on three benchmark datasets demonstrated that DPGNN outperformed the strong baselines.

Acknowledgements

We thank the anonymous reviewers for their valuable suggestions to improve the quality of this work. This work was partially supported by the National Key R&D Program of China (No. 2019YFB2102500), the National Natural Science Foundation of China (62006062, 62176076), Shenzhen Foundational Research Funding (JCYJ20200109113441941, JCYJ20210324115614039), The Major Key Project of PCL2021A06, Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (2022B1212010005).

Limitations

To better understand the limitations of the proposed model, we carry out an analysis of the errors made by DPGNN. Specifically, we randomly select 100 instances that are incorrectly predicted by DPGNN and summarize the primary types of error. The first type of error is caused by failing to classify ACs that contain latent opinions or require deep comprehension. For example, for an AC pair “Affirmative action is good public policy.” and “Predominantly black schools offer fewer AP classes.” DPGNN tends to align “good public policy” with “fewer AP classes”, resulting in attack relation which is wrongly predicted.

The second error category is caused by vague words. For example, DPGNN cannot correctly predict the argumentation relation between the AC pair High speed rail development is generally good policy.” and ”Upgrading existing lines is an ineffective solution.”, This may be because the context information is not sufficient enough such that DPGNN cannot capture the opposite semantic between ”High speed rail development” and ”Upgrading existing lines”.

Third, another error category occurs when the AC pair exists with multiple aligned antonyms. The argumentation relation is misled by multiple aligned antonyms between the AC pair. For example, the argumentation relation of the AC pair “Free trade and economic globalization is good for the world” and “Protectionism is discriminatory” is wrongly predicted as an attack by considering the two antonymous alignments between “Free trade and economic globalization” and “Protectionism”, and between “good” and “discriminatory”. It suggests that certain alignment method needs to be devised in the future so as to better infer argumentation relation. For example, we may leverage a graph neural network over the AC-specific node representations to guide the learning of relation-specific features.

References

Jianzhu Bao, Chuang Fan, Jipeng Wu, Yixue Dang, Jiachen Du, and Ruifeng Xu. 2021. A neural transition-based model for argumentation mining. In *Proceed-*

ings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 6354–6364.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. "what do neural machine translation models learn about morphology? In *ACL*.

Elena Cabrio and Serena Villata. 2012. Generating abstract arguments: A natural language approach. In *COMMA*, pages 454–461.

Di Chen, Jiachen Du, Lidong Bing, and Ruifeng Xu. 2018. Hybrid neural attention for agreement/disagreement inference in online debates. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 665–670.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

Oana Cocarascu and Francesca Toni. 2017. Identifying attack and support argumentative relations using deep learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1374–1379.

Junqi Dai, Hang Yan, Tianxiang Sun, Pengfei Liu, and Xipeng Qiu. 2021. Does syntax matter? a strong baseline for aspect-based sentiment analysis with roberta. In *EMNLP*.

Andrea Galassi, Marco Lippi, and Paolo Torroni. 2018. Argumentative link prediction using residual networks and multi-objective learning. In *Proceedings of the 5th Workshop on Argument Mining*, pages 1–10.

Debela Gemechu and Chris Reed. 2019. Decompositional argument mining: A general purpose approach for argument graph construction. In *57th Annual Meeting of the Association for Computational Linguistics*, pages 516–526. Association for Computational Linguistics.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. In *NAACL*.

Kuo-Yu Huang, Hen-Hsen Huang, and Hsin-Hsi Chen. 2021. Hargan: Heterogeneous argument attention network for persuasiveness prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13045–13054.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*.

Yohan Jo, Seojin Bang, Emaad Manzoor, Eduard Hovy, and Chris Reed. 2020. Detecting attackable sentences in arguments. *arXiv preprint arXiv:2010.02660*.

Yohan Jo, Seojin Bang, Chris Reed, and Eduard Hovy. 2021. Classifying argumentative relations using logical mechanisms and argumentation schemes. *Transactions of the Association for Computational Linguistics*, 9:721–739.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

John Lawrence and Chris Reed. 2020. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818.

Bin Liang, Yonghao Fu, Lin Gui, Min Yang, Jiachen Du, Yulan He, and Ruifeng Xu. 2021a. Target-adaptive graph for cross-target stance detection. In *Proceedings of the Web Conference 2021*, pages 3453–3464.

Bin Liang, Hang Su, Lin Gui, Erik Cambria, and Ruifeng Xu. 2022. Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks. *Knowledge-Based Systems*, 235:107643.

Bin Liang, Rongdi Yin, Jiachen Du, Lin Gui, Yulan He, Min Yang, and Ruifeng Xu. 2021b. Embedding refinement framework for targeted aspect-based sentiment analysis. *IEEE Transactions on Affective Computing*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. *arXiv preprint arXiv:1709.00893*.

Gaku Morio, Hiroaki Ozaki, Terufumi Morishita, Yuta Koreeda, and Kohsuke Yanai. 2020. Towards better non-tree argument mining: Proposition-level biaffine parsing with task-specific parameterization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3259–3266.

Raquel Mochales Palau and Marie-Francine Moens. 2009. Argumentation mining: the detection, classification and structure of arguments in text. In *Proceedings of the 12th international conference on artificial intelligence and law*, pages 98–107.

Debjit Paul, Juri Opitz, Maria Becker, Jonathan Kobbe, Graeme Hirst, and Anette Frank. 2020. Argumentative relation classification with background knowledge. In *Computational Models of Argument*, pages 319–330. IOS Press.

Andreas Peldszus. 2014. Towards segment-based recognition of argumentation structure in short texts. In *Proceedings of the First Workshop on Argumentation Mining*, pages 88–97.

Andreas Peldszus and Manfred Stede. 2015. Towards detecting counter-considerations in text. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 104–109.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B. Viégas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of BERT. In *NeurIPS*.

Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018. Improved semantic-aware network embedding with fine-grained word alignment. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1829–1838.

Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659.

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

Lihan Wang, Bowen Qin, Binyuan Hui, Bowen Li, Min Yang, Bailin Wang, Binhua Li, Jian Sun, Fei Huang, Luo Si, et al. 2022. Proton: Probing schema linking information from pre-trained language models for text-to-sql parsing. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1889–1898.

Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The world wide web conference*, pages 2022–2032.

Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. 2020. Perturbed masking: Parameter-free probing for analyzing and interpreting bert. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4166–4176.

Adam Wyner, Raquel Mochales-Palau, Marie-Francine Moens, and David Milward. 2010. Approaches to text mining arguments from legal cases. In *Semantic processing of legal texts*, pages 60–79. Springer.

Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Reasoning over semantic-level graph for fact checking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6170–6180.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [mask]: Learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5017–5033.