PAR: Political Actor Representation Learning with Social Context and Expert Knowledge

Shangbin Feng♣♦  Zhaoxuan Tan♣  Zilong Chen♣♦  Ningnan Wang♣
Peisheng Yu♣♦  Qinghua Zheng♣  Xiaojun Chang♦  Minnan Luo♣♦
Xi’an Jiaotong University♣, University of Washington♣, Tsinghua University♦
University of California San Diego♦, University of Technology Sydney♦
contact: shangbin@cs.washington.edu

Abstract

Modeling the ideological perspectives of political actors is an essential task in computational political science with applications in many downstream tasks. Existing approaches are generally limited to textual data and voting records, while they neglect the rich social context and valuable expert knowledge for holistic ideological analysis. In this paper, we propose PAR, a Political Actor Representation learning framework that jointly leverages social context and expert knowledge. Specifically, we retrieve and extract factual statements about legislators to leverage social context information. We then construct a heterogeneous information network to incorporate social context and use relational graph neural networks to learn legislator representations. Finally, we train PAR with three objectives to align representation learning with expert knowledge, model ideological stance consistency, and simulate the echo chamber phenomenon. Extensive experiments demonstrate that PAR is better at augmenting political text understanding and successfully advances the state-of-the-art in political perspective detection and roll call vote prediction. Further analysis proves that PAR learns representations that reflect the political reality and provide new insights into political behavior.

1 Introduction

Modeling the perspectives of political actors has applications in various downstream tasks such as roll call vote prediction (Mou et al., 2021) and political perspective detection (Feng et al., 2021). Existing approaches generally focus on voting records or textual information of political actors to induce their stances. Ideal point model (Clinton et al., 2004) is one of the most widely used approach for vote-based analysis, while later works enhance the ideal point model (Kraft et al., 2016; Gu et al., 2014; Gerrish and Blei, 2011) and yield promising results on the task of roll call vote prediction. For text-based methods, text analysis techniques are combined with textual information in social media posts (Li and Goldwasser, 2019), Wikipedia pages (Feng et al., 2021), legislative text (Mou et al., 2021) and news articles (Li and Goldwasser, 2021) to enrich the perspective analysis process.

However, existing methods fail to incorporate the rich social context and valuable expert knowledge of political actors. As illustrated in Figure 1, social context information such as home state and party affiliation serves as background knowledge and helps connect different political actors (Yang et al., 2021). These social context facts about political actors also differentiate them and indicate their ideological stances. In addition, expert knowledge from political think tanks provides valuable insights and helps to anchor the perspective analysis process. As a result, political actor representation learning should be guided by domain expertise to facilitate downstream tasks in computational political science. That being said, social context and expert knowledge should be incorporated in modeling legislators to ensure a holistic evaluation.

In light of these challenges, we propose PAR, a legislator representation learning framework that jointly leverages social context and expert knowledge. We firstly collect a dataset of political actors by retrieving and extracting social context information from their Wikipedia homepages and adapting expert knowledge from two political think...
tanks AFL-CIO\(^1\) and Heritage Action\(^2\). After that, we construct a heterogeneous information network to model social context information and adopt relational graph neural networks for representation learning. Finally, we train the framework with three training objectives to leverage expert knowledge, model social and political phenomena, and learn representations of political actors in the process. We evaluate PAR on two computational political science tasks and examine the political behaviour of learned representations. Our main contributions are summarized as follows:

- We propose PAR, a graph-based approach to learn legislator representations with three training objectives, which aligns representation learning with expert knowledge, ensures stance consistency, and models the echo chamber phenomenon in socio-economic systems.

- Extensive experiments demonstrate that PAR advances the state-of-the-art on two computational political science tasks, namely political perspective detection and roll call vote prediction.

- Further analysis shows that PAR learns representations that reflect the ideological preferences and political behavior of political actors such as legislators, governors, and states. In addition, PAR provides interesting insights into the contemporary political reality.

2 Related Work

The ideological perspectives of political actors play an essential role in their individual behavior and adds up to influence the overall legislative process (Freeden, 2006; Bamman et al., 2012; Wilkerson and Casas, 2017). Political scientists first explored to quantitatively model political actors based on their voting behaviour. Ideal point model (Clinton et al., 2004) is one of the earliest approach to leverage voting records to analyze their perspectives. It projects political actors and legislation onto one-dimensional spaces and measure distances. Many works later extended the ideal point model. Gerrish and Blei (2011) leverages bill content to infer legislator perspectives. Gu et al. (2014) introduces topic factorization to model voting behaviour on different issues to establish a fine-grained approach. Kraft et al. (2016) models legislators with multidimensional ideal vectors to analyze voting records. Mohammad et al. (2017) and Küçük and Can (2018) use SVM and handcrafted text features to augment the analysis.

In addition to voting, textual data such as speeches and public statements are also leveraged to model political perspectives (Evans et al., 2007; Thomas et al., 2006; Hasan and Ng, 2013; Zhang et al., 2022; Sinno et al., 2022; Liu et al., 2022; Davoodi et al., 2022; Alkiek et al., 2022; Dayanik et al., 2022; Pujari and Goldwasser, 2021; Villegas et al., 2021; Li et al., 2021; Sen et al., 2020; Baly et al., 2020). Volkova et al. (2014) leverage message streams to inference users political preference. Johnson and Goldwasser (2016) propose to better understand political stances by analyzing politicians’ tweet and their temporal activities. Li and Goldwasser (2019) propose to analyze Twitter posts to better understand news stances. Prakash and Tayyar Madabushi (2020) and Kawintiranon and Singh (2021) leverage pre-trained language model for stance prediction. Augenstein et al. (2016) uses conditional LSTM to encode tweets for stance detection. Furthermore, CNN (Wei et al., 2016) and hierarchical attention networks (Sun et al., 2018a) are adopted for stance detection. Darwish et al. (2020) proposes an unsupervised framework for user representation learning and stance detection. Yang et al. (2021) proposes to jointly model legislators and legislations for vote prediction. Feng et al. (2021) introduces Wikipedia corpus and constructs knowledge graphs to facilitate perspective detection. Mou et al. (2021) proposes to leverage tweets, hashtags and legislative text to grasp the full picture of the political discourse. Li and Goldwasser (2021) designs pre-training tasks with social and linguistic information to augment political analysis. However, these vote and text-based methods fail to leverage the rich social context of political actors and the valuable expert knowledge of political think tanks. In this paper, we aim to incorporate social context and expert knowledge while focusing on learning representations of political actors to model their perspectives and facilitate downstream tasks.

3 Methodology

Figure 2 presents an overview of PAR (Political Actor Represetation learning). We firstly collect a dataset of political actors from Wikipedia and political think tanks. We then construct a heterogeneous

\(^1\)https://aflcio.org/
\(^2\)https://heritageaction.com/
information network to jointly model political actors and social context information. Finally, we learn graph representations with gated relational graph convolutional networks (gated R-GCN) and train the framework with three different objectives to leverage expert knowledge and model various socio-political phenomena.

3.1 Data Collection
We firstly collect a dataset about political actors in the United States that were active in the past decade. For social context information, we retrieve the list of senators and congresspersons from the 114th congress to the 117th congress. We then retrieve their Wikipedia pages and extract these named entities: presidents, senators, congresspersons, governors, states, political parties, supreme court justices, government institutions, and office terms (117th congress etc.). In this way, we obtain 1,069 social and political entities. Based on these entities, we identify five types of relations: party affiliation, home state, political office, term in office, and appoint relationships. In this way, we obtain 9,248 heterogeneous edges. For expert knowledge, we use legislator scoreboards at AFL-CIO and Heritage Action, two political think tanks that lie in the opposite ends of the ideological spectrum. Specifically, we retrieve and extract each legislator’s score in each office term. In this way, we obtain 777 scores from AFL-CIO and 679 scores from Heritage Action. We consolidate the collected social context and expert knowledge to serve as the dataset in our experiments. Although we focus on political actors in the United States in this paper, our data collection process is also applicable for other countries and time ranges.

3.2 Graph Construction
To better model the interactions between political entities and their shared social context, we propose to construct a heterogeneous information network (HIN) from the dataset.

3.2.1 Heterogeneous Nodes
Based on the collected dataset, we select diversified entities that are essential factors in modeling the political process. Specifically, we use eight types of nodes to represent political actors and diversified social context entities.

$N_1$: Office Terms. We use four nodes to represent the 114th, 115th, 116th, 117th congress spanning from 2015 to 2022. We use these nodes to model different political scenarios and could be similarly extended to other time periods.

$N_2$: Legislators. We retrieve senators and congresspersons from the 114th to 117th congress and use one node to represent each distinct legislator.

$N_3$: Presidents. President is the highest elected office in the United States. We use three nodes to represent President Biden, Trump and Obama to match with the time range in $N_1$.

$N_4$: Governors. State and local politics are also essential in analyzing the political process. We use

---

3.1 Data Collection

We firstly collect a dataset about political actors in the United States that were active in the past decade. For social context information, we retrieve the list of senators and congresspersons from the 114th congress to the 117th congress. We then retrieve their Wikipedia pages and extract these named entities: presidents, senators, congresspersons, governors, states, political parties, supreme court justices, government institutions, and office terms (117th congress etc.). In this way, we obtain 1,069 social and political entities. Based on these entities, we identify five types of relations: party affiliation, home state, political office, term in office, and appoint relationships. In this way, we obtain 9,248 heterogeneous edges. For expert knowledge, we use legislator scoreboards at AFL-CIO and Heritage Action, two political think tanks that lie in the opposite ends of the ideological spectrum. Specifically, we retrieve and extract each legislator’s score in each office term. In this way, we obtain 777 scores from AFL-CIO and 679 scores from Heritage Action. We consolidate the collected social context and expert knowledge to serve as the dataset in our experiments. Although we focus on political actors in the United States in this paper, our data collection process is also applicable for other countries and time ranges.

3.2 Graph Construction

To better model the interactions between political entities and their shared social context, we propose to construct a heterogeneous information network (HIN) from the dataset.

3.2.1 Heterogeneous Nodes

Based on the collected dataset, we select diversified entities that are essential factors in modeling the political process. Specifically, we use eight types of nodes to represent political actors and diversified social context entities.

$N_1$: Office Terms. We use four nodes to represent the 114th, 115th, 116th, 117th congress spanning from 2015 to 2022. We use these nodes to model different political scenarios and could be similarly extended to other time periods.

$N_2$: Legislators. We retrieve senators and congresspersons from the 114th to 117th congress and use one node to represent each distinct legislator.

$N_3$: Presidents. President is the highest elected office in the United States. We use three nodes to represent President Biden, Trump and Obama to match with the time range in $N_1$.

$N_4$: Governors. State and local politics are also essential in analyzing the political process. We use
one node to represent each distinct governor of 50 states within the time range of N1.

N5: States. The home state of political actors is often an important indicator and helps connect different individuals. We use one node to represent each state in the United States.

N6: Government Institutions. We use five nodes to represent the white house, senate, house of representatives, supreme court and governorship. These nodes enable our HIN to separate different political actors based on the office they hold.

N7: Supreme Court Justices. Supreme court justices are nominated by presidents and approved by senators, which helps connect different types of political actors. We use one node to represent each supreme court justice within the time range of N1.

N8: Political Parties. We use two nodes to represent the Republican Party and the Democratic Party in the United States.

For node features, we use pre-trained RoBERTa (Liu et al., 2019) to encode the first paragraph of Wikipedia pages and average all tokens.

3.2.2 Heterogeneous Relations
Based on N1 to N8, we extract five types of informative interactions between entities to complete the HIN structure. Specifically, we use five types of heterogeneous relations to connect different nodes and construct our political actor HIN.

R1: Party Affiliation. We connect political actors and their affiliated political party with R1:

\[ R1 = (N2 \cup N3 \cup N4) \times N8 \]  

R2: Home State. We connect political actors with their home states with R2:

\[ R2 = (N2 \cup N3 \cup N4 \cup N7) \times N5 \]  

R3: Hold Office. We connect political actors with the political office they hold with R3:

\[ R3 = (N2 \cup N3 \cup N4 \cup N7) \times N6 \]  

R4: Time in Office. If a political actor holds office during one of the time stamps in N1, we connect them with R4:

\[ R4 = (N2 \cup N3 \cup N4 \cup N7) \times N1 \]  

R5: Appoint. Besides from being elected, certain political actors are appointed by others. We denote this relation with R5:

\[ R5 = (N3 \times N7) \cup (N4 \times N2) \]  

3.3 Representation Learning
Since nodes represent political actors, we learn node-level representations with gated R-GCN to jointly leverage social context and expert knowledge. Let \( E = \{e_1, \cdots, e_n\} \) be \( n \) entities in the HIN and \( v_i \) be the initial features of entity \( e_i \). Let \( R \) be the heterogeneous relation set and \( N_r(e_i) \) be entity \( e_i \)'s neighborhood with regard to relation type \( r \). We firstly transform \( v_i \) to serve as the input of graph neural networks:

\[ x_i^{(0)} = \phi(W_l \cdot v_i + b_l) \]  

where \( \phi \) is leaky-relu, \( W_l \) and \( b_l \) are learnable parameters. We then propagate entity messages and aggregate them with gated R-GCN. For the \( l \)-th layer of gated R-GCN:

\[ u_i^{(l)} = \sum_{r \in R} \sum_{j \in N_r(e_i)} f_r(x_j^{(l-1)}) + f_s(x_i^{(l-1)}) \]  

where \( f_s \) and \( f_r \) are parameterized linear layers for self loops and edges of relation \( r \). \( u_i^{(l)} \) is the hidden representation for entity \( e_i \) at layer \( l \). We then calculate gate levels:

\[ g_i^{(l)} = \sigma(W_G \cdot [u_i^{(l)}, x_i^{(l-1)}] + b_G) \]  

where \( W_G \) and \( b_G \) are learnable parameters, \( \sigma(\cdot) \) denotes the sigmoid function and \( [\cdot, \cdot] \) denotes the concatenation operation. We then apply the gate mechanism to \( u_i^{(l)} \) and \( x_i^{(l-1)} \):

\[ x_i^{(l)} = \tanh(u_i^{(l)}) \odot g_i^{(l)} + x_i^{(l-1)} \odot (1 - g_i^{(l)}) \]  

where \( \odot \) is the Hadamard product. After \( L \) layer(s) of gated R-GCN, we obtain node features \( \{x_1^{(L)}, \cdots, x_n^{(L)}\} \) and the nodes representing political actors are extracted as learned representations.

3.4 Model Training
We propose to train PAR with a combination of three training objectives, which aligns representation learning with expert knowledge, ensures stance consistency and simulates the echo chamber phenomenon. The overall loss function of PAR is as follows:

\[ L = \lambda_1 L_1 + \lambda_2 L_2 + \lambda_3 L_3 + \lambda_4 \sum_{w \in \theta} ||w||^2 \]  

where \( \lambda_i \) is the weight of loss \( L_i \) and \( \theta \) are learnable parameters in PAR. We then present the motivation and detail of each loss function \( L_1, L_2 \) and \( L_3 \).
3.4.1 Objective 1: Expert Knowledge
The expert knowledge objective aims to align the learned representations with expert knowledge from political think tanks. We use the learned representations of political actors to predict their liberal and conservative stances, which are adapted from AFL-CIO and Heritage Action. Specifically:

\[ l_i = \text{softmax}(W_L \cdot x_i^{(L)} + b_L) \]
\[ c_i = \text{softmax}(W_C \cdot x_i^{(L)} + b_C) \]

where \( l_i \) and \( c_i \) are predicted stances towards liberal and conservative values, \( W_L, b_L, W_C \) and \( b_C \) are learnable parameters. Let \( E_L \) and \( E_C \) be the training set of AFL-CIO and Heritage Action scores, \( \hat{l}_i \) and \( \hat{c}_i \) denote the ground-truth among \( D \) possible labels. We calculate the expert knowledge loss:

\[
L_1 = - \sum_{e_i \in E_L} \sum_{d=1}^{D} \hat{l}_{id} \log(l_{id}) - \sum_{e_i \in E_C} \sum_{d=1}^{D} \hat{c}_{id} \log(c_{id})
\]

(12)

\( L_1 \) enables PAR to align learned representations with expert knowledge from political think tanks.

3.4.2 Objective 2: Stance Consistency
The stance consistency objective is motivated by the fact that liberalism and conservatism are opposite ideologies, thus individuals often take inversely correlated stances towards them. We firstly speculate entities’ stance towards the opposite ideology by taking the opposite of the predicted stance:

\[
\hat{l}_i = \psi(D - \text{argmax}(c_i)), \quad \hat{c}_i = \psi(D - \text{argmax}(l_i))
\]

(13)

where \( \psi \) is the one-hot encoder, \( \text{argmax}(\cdot) \) calculates the vector index with the largest value, \( D \) is the number of stance labels, \( \hat{l}_i \) and \( \hat{c}_i \) are labels derived with stance consistency. We calculate the loss function \( L_2 \) measuring stance consistency:

\[
L_2 = - \sum_{e_i \in E} \sum_{d=1}^{D} (\hat{l}_{id} \log(l_{id}) + \hat{c}_{id} \log(c_{id}))
\]

(14)

where \( E = E_L \cap E_C \). As a result, the loss function \( L_2 \) enables PAR to ensure ideological stance consistency among political actors.

3.4.3 Objective 3: Echo Chamber
The echo chamber objective is motivated by the echo chamber phenomenon (Jamieson and Cappella, 2008; Barberá et al., 2015), where social entities tend to reinforce their narratives by forming small and closely connected interaction circles. We simulate echo chambers by assuming that neighboring nodes in the HIN have similar representations while non-neighboring nodes have different representations. We firstly define the positive and negative neighborhood of entity \( e_i \):

\[
P_{e_i} = \{ e \mid \exists r \in R \text{ s.t. } e \in N_r(e_i) \}
\]
\[
N_{e_i} = \{ e \mid \forall r \in R \text{ s.t. } e \notin N_r(e_i) \}
\]

(15)

We then calculate the echo chamber loss:

\[
L_3 = - \sum_{e_i \in E} \sum_{e_j \in P_{e_i}} \log(\sigma(x_i^{(L)^T} x_j^{(L)})) + Q \cdot \sum_{e_i \in E} \sum_{e_j \in N_{e_i}} \log(\sigma(-x_i^{(L)^T} x_j^{(L)}))
\]

(16)

where \( Q \) denotes the weight for negative samples. \( L_3 \) enables PAR to model the echo chamber phenomenon in real-world socio-economic systems.

4 Experiment
After learning political actor representations with PAR, we study whether they are effective in computational political science and reveal real-world political behavior. Specifically, we test out PAR on political perspective detection and roll call vote prediction, two political text understanding tasks that emphasizes political actor modeling. We leverage PAR in these tasks to examine whether it could contribute to the better understanding of political text. We then study the learned representations of PAR and whether they provide new insights into real-world politics.

4.1 Political Perspective Detection
Political perspective detection aims to detect stances in text such as public statements and news articles, which generally mention many political actors to provide context and present arguments. Existing methods model political actors with masked entity models (Li and Goldwasser, 2021) and knowledge graph embedding techniques (Feng et al., 2021). We examine whether PAR is more effective than these models and consequently improve political perspective detection.

4.1.1 Datasets
We follow previous works (Li and Goldwasser, 2021; Feng et al., 2021) and adopt two political perspective detection benchmarks: SemEval and Allsides. SemEval (Kiesel et al., 2019) aims to
<table>
<thead>
<tr>
<th>Method</th>
<th>Setting</th>
<th>SemEval Acc</th>
<th>MaF</th>
<th>AllSides Acc</th>
<th>MaF</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td>GloVe</td>
<td>79.63</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>ELMo</td>
<td>84.04</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>HLSTM</td>
<td>GloVe</td>
<td>81.58</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>ELMo</td>
<td>83.28</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Embed</td>
<td>81.71</td>
<td>N/A</td>
<td>76.45</td>
<td>74.95</td>
</tr>
<tr>
<td></td>
<td>Output</td>
<td>81.25</td>
<td>N/A</td>
<td>76.66</td>
<td>75.39</td>
</tr>
<tr>
<td>BERT</td>
<td>base</td>
<td>84.03</td>
<td>82.60</td>
<td>81.55</td>
<td>80.13</td>
</tr>
<tr>
<td>MAN</td>
<td>GloVe</td>
<td>81.58</td>
<td>79.29</td>
<td>78.29</td>
<td>76.96</td>
</tr>
<tr>
<td></td>
<td>ELMo</td>
<td>84.66</td>
<td>83.09</td>
<td>81.41</td>
<td>80.44</td>
</tr>
<tr>
<td></td>
<td>Ensemble</td>
<td>86.21</td>
<td>84.33</td>
<td>85.00</td>
<td>84.25</td>
</tr>
<tr>
<td>KGAP</td>
<td>TransE</td>
<td>89.56</td>
<td>84.94</td>
<td>86.02</td>
<td>85.52</td>
</tr>
<tr>
<td></td>
<td>TransR</td>
<td>88.54</td>
<td>83.45</td>
<td>85.15</td>
<td>84.61</td>
</tr>
<tr>
<td></td>
<td>DistMult</td>
<td>88.51</td>
<td>83.63</td>
<td>84.47</td>
<td>83.90</td>
</tr>
<tr>
<td></td>
<td>HolE</td>
<td>88.85</td>
<td>83.68</td>
<td>84.78</td>
<td>84.24</td>
</tr>
<tr>
<td></td>
<td>RotatE</td>
<td>88.84</td>
<td>84.04</td>
<td>85.61</td>
<td>85.11</td>
</tr>
</tbody>
</table>

Table 1: Political perspective detection performance on two benchmark datasets. Acc and MaF denote accuracy and macro-averaged F1-score. N/A indicates that the result is not reported in previous works.

identify whether a news article follows hyperpartisan argumentation. We follow the 10-fold cross validation setting and the exact same folds established in Li and Goldwasser (2019) so that the results are directly comparable. Allsides (Li and Goldwasser, 2019) provides left, center, or right labels for three-way classification. We follow the 3-fold cross validation setting and the exact same folds established in Li and Goldwasser (2019) so that the results are directly comparable.

4.1.2 Baselines
We compare PAR with competitive baselines:

- **CNN** (Jiang et al., 2019) achieves the best performance in the SemEval 2019 Task 4 contest (Kiesel et al., 2019). It uses convolutional neural networks along with word embeddings GloVe (Jiang et al., 2019) and ELMo (Peters et al., 2018) for perspective detection.

- **HLSTM** (Yang et al., 2016) is short for hierarchical long short-term memory networks. Li and Goldwasser (2019) combines it with GloVe and ELMo word embeddings. Li and Goldwasser (2021) leverages Wikipedia2Vec (Yamada et al., 2020) and masked entity models while using different concatenation strategies (HLSTM_Embed and HLSTM_Output).

<table>
<thead>
<tr>
<th>Method</th>
<th>Setting</th>
<th>random</th>
<th>time-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>majority</td>
<td>77.48</td>
<td>77.40</td>
<td></td>
</tr>
<tr>
<td>ideal-point-wf</td>
<td>85.37</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>ideal-point-tfidf</td>
<td>86.48</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>ideal-vector</td>
<td>87.35</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>CNN</td>
<td>87.28</td>
<td>81.97</td>
<td></td>
</tr>
<tr>
<td>CNN+meta</td>
<td>88.02</td>
<td>84.30</td>
<td></td>
</tr>
<tr>
<td>LSTM+GCN</td>
<td>88.41</td>
<td>85.82</td>
<td></td>
</tr>
<tr>
<td>Vote</td>
<td>90.22</td>
<td>89.76</td>
<td></td>
</tr>
<tr>
<td>RoBERTa</td>
<td>87.59</td>
<td>87.56</td>
<td></td>
</tr>
<tr>
<td>TransE</td>
<td>82.70</td>
<td>80.06</td>
<td></td>
</tr>
<tr>
<td>PAR</td>
<td>90.33</td>
<td>89.92</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Roll call vote prediction performance (accuracy) with two experiment settings. N/A indicates that the result is not reported in previous works.

- **BERT** (Devlin et al., 2019) is fine-tuned on the task of political perspective detection.

- **MAN** (Li and Goldwasser, 2021) leverages social and linguistic information for pre-training a BERT-based model and fine-tune on the task of political perspective detection.

- **KGAP** (Feng et al., 2021) models legislators with RoBERTa and knowledge graph embeddings: TransE (Bordes et al., 2013), TransR (Lin et al., 2015), DistMult (Yang et al., 2015), HolE (Nickel et al., 2016), and RotatE (Sun et al., 2018b). It then constructs document graphs with text and legislators as nodes and uses GNNs for stance detection.

- **KGAP_PAR**: To examine whether PAR is effective in political perspective detection, we replace the knowledge graph embeddings in KGAP with political actor representations learned with PAR.

4.1.3 Results
We evaluate PAR and baselines and present model performance in Table 1, which demonstrates that PAR achieves state-of-the-art performance on both datasets. The fact that PAR outperforms MAN and KCD, two methods that also take political actors into account, shows that PAR learns high-quality representations that results in performance gains.

4.2 Roll Call Vote Prediction
Roll call vote prediction aims to predict whether a legislator will vote in favor or against a specific legislation. Ideal points (Gerrish and Blei, 2011), ideal vectors (Kraft et al., 2016), and Mou et al. (2021)
have been proposed to model ideological preferences. We examine whether PAR is more effective than these models in roll call vote prediction.

### 4.2.1 Datasets

We adopt the datasets and settings proposed in Mou et al. (2021) to evaluate PAR and competitive baselines. Specifically, for the random setting, voting records in the 114th and 115th congresses are randomly split into 6:2:2 for training, validation, and testing. For the time-based setting, the voting records in the 114th congress is split into 8:2 for training and validation, while the voting records in the 115th congress is used for testing.

### 4.2.2 Baselines

We compare PAR with competitive baselines:

- **majority** predicts all votes as *yea*.

- **ideal-point-wf** and **ideal-point-tfidf** (Gerrish and Blei, 2011) adopt word frequency and TFIDF of legislation text as features and leverage the ideal point model for vote prediction.

- **ideal-vector** (Kraft et al., 2016) learns distributed representations with legislator and bill text.

- **CNN** (Kornilova et al., 2018) encodes bill text with CNNs. On top of that, **CNN+meta** introduces metadata of bill sponsorship to the process.

- **LSTM+GCN** (Yang et al., 2021) leverages LSTM and GCN to jointly update representations of legislations and legislators.

- **Vote** (Mou et al., 2021) proposes to align statements on social networks with voting records.

- **RoBERTa** and **TransE** use RoBERTa (Liu et al., 2019) encoding of legislator description on Wikipedia or TransE (Bordes et al., 2013) embeddings for legislator representation learning. They are then concatenated with RoBERTa encoded bill text for vote prediction.

- **PAR** concatenates political actor representations learned with PAR and RoBERTa encoded legislator text for roll call vote prediction.

### 4.2.3 Results

We evaluate PAR and competitive baselines on roll call vote prediction and present model performance in Table 2. PAR achieves state-of-the-art performance, outperforming existing baselines that model political actors in different ways. As a result, PAR learns high-quality representations of political actors that provide political knowledge and augment the vote prediction process.

### 4.3 Political Findings of PAR

PAR learns political actor representations with social context and expert knowledge, which has been proven effective in political perspective detection and roll call vote prediction. We further examine whether PAR provides new insights into the political behavior of legislators, states, and governors.

#### 4.3.1 Legislators and PAR

To examine whether legislator representations learned with PAR align well with different social
and political factors, we adopt t-sne (Maaten and Hinton, 2008) to illustrate our learned representations of political actors in Figure 3.

Social context. Figure 3 (a) and (b) illustrate the correlation between learned representations and social context factors such as party affiliation and home state. It is illustrated that legislators from the same party and the same home state tends to be similar in the representation space. In addition, we calculate the DBI scores (Davies and Bouldin, 1979) to quantitatively analyze the collocation. In conclusion, the learned representations successfully reflect these social context information.

Congressional caucus. Figure 3 (c) and (d) demonstrate the correlation between learned representations and major congressional caucuses in both parties. The progressive caucus in the democratic party and the freedom caucus in the republican party are often viewed as more ideological wings of the party. Both the illustration and the DBI scores suggest little collocation among different caucuses. As a result, PAR reveals that despite widening ideological gaps between party factions (Cohen et al., 2016), inter-party differences still outweigh intra-party differences in contemporary U.S. politics.

Voting records. Figure 3 (e), (f), (g) and (h) illustrate how often does a legislator vote for or against the sitting president. We retrieve this information from FiveThirtyEight5 and illustrate voting records with color gradients. "Trump scores" and "Biden scores" indicate what percentage out of all votes did a legislator vote with or against the official stance of the president. As a result, our learned representations of legislators in the 115th and 116th congress correlate well with their voting records, while the 117th congress might have not hold enough votes for an accurate categorization6.

4.3.2 States and PAR
Political commentary often uses "blue state", "red state", and "swing state" to describe the ideological preference of states in the U.S. We examine the ideological scores of states learned by PAR ($l_i$ and $c_i$ in Equation (11)) and compare them with results in the 2020 U.S. presidential election in Figure 4. It is illustrated that PAR stance predictions highly correlate with the 2020 presidential election results. In addition, PAR reveals that Pennsylvania and North Carolina, two traditional swing states, are actually more partisan than expected. PAR also suggests that Georgia is the most electorally competitive state in the United States and we will continue to monitor this conclusion in future elections.

4.3.3 Governors and PAR
Political experts typically study and evaluate the stances of presidents and federal legislators, while state-level officials such as governors are also essential in governance and policy making (Beyle, 1988). PAR complements the understanding of state-level politics by learning ideological scores for governors. Specifically, we use $l_i$ and $c_i$ in Equation (11) to evaluate the ideological position of governors and present PAR’s predictions of governor stances in all 50 U.S. states in Figure 57. It is no surprise that governors from partisan strongholds such as California and Utah hold firm stances. Conventional wisdom often assumes that in order to

5https://fivethirtyeight.com/

6Voting records of the 117th congress is collected in June 2021, only five months into its term.

7This figure is created with the help of mapchart.net.
win "swing states" or electorally challenging races, one has to compromise their ideological stances and lean to the center. However, PAR reveals exceptions to this rule, such as Andy Beshear (D-KY) and Ron DeSantis (R-FL), which is also suggested in political commentary (Jr., 2020; Graham, 2022).

5 Conclusion

In this paper, we present PAR, a framework to learn representations of political actors with social context and expert knowledge. Specifically, we retrieve social context information from Wikipedia and expert knowledge from political think tanks, construct a HIN to model legislators and learn representations with gated R-GCNs and three training objectives. Extensive experiments demonstrate that PAR advances the state-of-the-art in political perspective detection and roll call vote prediction. PAR further provides novel insights into political actors through its representation learning process.

6 Limitations

- PAR proposes to learn representations for political actors with the help of social context and expert knowledge. Though it achieved great performance on several tasks and provided interesting political insights, it might reinforce political stereotypes, such as people from "red states" are often assumed to be more conservative. We leave for future work how to mitigate the potential bias in political actor representations.

- In this paper, we focus on political actors in the United States instead of other countries. However, our proposed framework could be easily extended to other scenarios by leveraging the Wikipedia pages of political actors for social context and political think tanks in other countries for expert knowledge.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (No. 2020AAA0108800), National Nature Science Foundation of China (No. 62192781, No. 62272374, No. 61872287, No. 62250009, No. 62137002), Innovative Research Group of the National Natural Science Foundation of China (61721002), Innovation Research Team of Ministry of Education (IRT_17R86), Project of China Knowledge Center for Engineering Science and Technology and Project of Chinese academy of engineering “The Online and Offline Mixed Educational Service System for ‘The Belt and Road’ Training in MOOC China”.

We would like to thank the reviewers and area chair for their constructive feedback. We would also like to thank all LUD lab members for our collaborative research environment.

References


Yujian Liu, Xinliang Frederick Zhang, David Wegsman, Nicholas Beauchamp, and Lu Wang. 2022. POLITICS: Pretraining with same-story article comparison for ideology prediction and stance detection. In Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, United States.


<table>
<thead>
<tr>
<th>Method</th>
<th>Text</th>
<th>Graph</th>
<th>Acc</th>
<th>MaF</th>
<th>MiF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear BoW</td>
<td>✓</td>
<td></td>
<td>68.49</td>
<td>40.00</td>
<td>68.53</td>
</tr>
<tr>
<td>Bias Features</td>
<td>✓</td>
<td></td>
<td>47.26</td>
<td>20.08</td>
<td>47.10</td>
</tr>
<tr>
<td>Glove</td>
<td>✓</td>
<td></td>
<td>52.05</td>
<td>26.94</td>
<td>52.01</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>✓</td>
<td></td>
<td>71.92</td>
<td>49.70</td>
<td>71.87</td>
</tr>
<tr>
<td>LongFormer</td>
<td>✓</td>
<td></td>
<td>68.49</td>
<td>42.27</td>
<td>68.56</td>
</tr>
<tr>
<td>GCN</td>
<td>✓</td>
<td>✓</td>
<td>74.66</td>
<td>54.16</td>
<td>74.46</td>
</tr>
<tr>
<td>GAT</td>
<td>✓</td>
<td>✓</td>
<td>78.08</td>
<td>55.82</td>
<td>78.17</td>
</tr>
<tr>
<td>GraphSAGE</td>
<td>✓</td>
<td>✓</td>
<td>75.34</td>
<td>51.39</td>
<td>75.43</td>
</tr>
<tr>
<td>TransformerConv</td>
<td>✓</td>
<td>✓</td>
<td>77.40</td>
<td>55.63</td>
<td>77.48</td>
</tr>
<tr>
<td>ResGatedConv</td>
<td>✓</td>
<td>✓</td>
<td>76.03</td>
<td>54.31</td>
<td>75.97</td>
</tr>
<tr>
<td>Ours</td>
<td>✓</td>
<td>✓</td>
<td>80.82</td>
<td>60.37</td>
<td>80.89</td>
</tr>
</tbody>
</table>

Table 3: Our model’s performance on the expert knowledge prediction task compared to various text and graph analysis baselines. Acc, MaF and MiF denote accuracy, macro and micro-averaged F1-score.

A Expert Knowledge Prediction

We propose to train PAR with three objectives, the first one $L_1$ being expert knowledge prediction, where the model learns to predict how liberal or conservative a given legislator is. To examine whether the PAR architecture successfully conducts expert knowledge prediction, we compare PAR with various text (Pedregosa et al., 2011; Horne et al., 2018; Pennington et al., 2014; Liu et al., 2019; Beltagy et al., 2020) and graph (Kipf and Welling, 2016; Veličković et al., 2018; Hamilton et al., 2017; Shi et al., 2020; Bresson and Laurent, 2017) analysis baselines. Table 3 presents model performance on the expert knowledge task. It is demonstrated that PAR achieves the best performance compared to various text and graph analysis baselines, suggesting our learned representations successfully reflect expert knowledge from political think tanks. Apart from that, graph-based models generally outperform text-based methods, which suggests that the constructed HIN is essential in modeling political actors.

B Ablation Study

PAR aims to learn representations of political actors with social context and expert knowledge as well as three training objectives. We conduct ablation study to examine their effect in the representation learning process. We report model performance on the expert knowledge prediction task and follow the same dataset splits as in Table 3.

B.1 Social Context

We use five types of heterogeneous relations $R_1$ to $R_5$ to connect different entities based on their social context. We randomly and gradually remove five types of social context edges in the constructed HIN and report model performance in Figure 6. It is illustrated that all relations but $R_4$ (Time in Office) significantly contributes to the overall performance. Besides, Figure 6 illustrates a great gap between 90% and 100% edges, suggesting the importance of a complete HIN structure.

B.2 Expert Knowledge

We learn legislator representations with the help of two political think tanks: AFL-CIO and Heritage Action. We retrieve their evaluation of political actors and construct the expert knowledge objective for training. To examine the effect of expert knowledge in our proposed approach, we gradually remove expert knowledge labels in $L_1$ and report model performance in Figure 7. It is illustrated that our performance drops with partial expert knowledge from either AFL-CIO or Heritage Action, which indicates that expert knowledge is essential in the representation learning process.

B.3 Training Objectives

We propose to train our framework with three objectives. To examine their effect, we train our method with different combinations of $L_1$, $L_2$ and $L_3$ and report performance in Table 4. Our model performs best with all three training objectives, proving the effectiveness of our loss function design. We further study the influence of loss weights $\lambda_1$, $\lambda_2$, $\lambda_3$ and $\lambda_4$. We fix $\lambda_1 = 1$ and $\lambda_4 = 10^{-5}$, present model performance under different settings of loss weights for auxiliary tasks $\lambda_2$ and $\lambda_3$ in Figure 8. It is illustrated that $0.2 \leq \lambda_2 \leq 0.3$ and
Figure 7: Ablation study of expert knowledge. We report accuracy on the expert knowledge prediction task.

<table>
<thead>
<tr>
<th>Loss Function(s)</th>
<th>Acc</th>
<th>MaF</th>
<th>MiF</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_1$ only</td>
<td>78.08</td>
<td>56.43</td>
<td>78.17</td>
</tr>
<tr>
<td>$L_1$ and $L_2$</td>
<td>79.45</td>
<td>55.97</td>
<td>79.52</td>
</tr>
<tr>
<td>$L_1$ and $L_3$</td>
<td>76.03</td>
<td>51.91</td>
<td>76.11</td>
</tr>
<tr>
<td>$L_1$, $L_2$, and $L_3$</td>
<td>80.82</td>
<td>60.37</td>
<td>80.89</td>
</tr>
</tbody>
</table>

Table 4: Ablation study of three training objectives.

0.01 $\leq \lambda_3 \leq 0.1$ would generally lead to an effective balance of three different training objectives.

B.4 Graph Learning

We adopted five relations $R_1$ to $R_5$ to connect eight types of entities $N_1$ to $N_8$, so that the graph is heterogeneous. To examine whether the graph heterogeneity contributes to model performance, we substitute gated R-GCNs with homogeneous GNNs such as GCN, GAT, and GraphSAGE. Table 5 shows mixed results, where PAR performs best with gated R-GCN while R-GCN does not outperform GraphSAGE.

C Error Analysis

We manually examined the part of the results in the roll call vote prediction task. Among the (legislator, bill) pairs where PAR made a wrong prediction, it is often the case that the legislator has voted across party lines. This suggests that more information about these legislators are required to achieve more fine-grained analysis on these borderline cases.

D Reproducibility Details

In this section, we provide additional details to facilitate reproducing our results and findings. We submit data and code as supplementary material and commit to make them publicly available upon acceptance to facilitate reproduction.

D.1 Hyperparameters

We present the hyperparameter settings of our proposed approach in Table 6. These hyperparameters are manually tuned. We follow these settings throughout the paper unless stated otherwise.

D.2 Implementation

We use pytorch (Paszke et al., 2019), pytorch lightning (Falcon and The PyTorch Lightning team, 2019), torch geometric (Fey and Lenssen, 2019), and the transformers library (Wolf et al., 2020) for implementation. All implemented codes are available as supplementary material.

D.3 Computation

Our proposed approach has a total of 2.0M learnable parameters with hyperparameters in Table 6. We train it on a Titan X GPU with 12GB memory. It takes approximately 0.6 GPU hours for training with hyperparameters in Table 6.
Table 6: Hyperparameters of our proposed approach.

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Value</th>
<th>Hyperparameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RoBERTa size</td>
<td>768</td>
<td>GNN size</td>
<td>512</td>
</tr>
<tr>
<td>optimizer</td>
<td>Adam</td>
<td>learning rate</td>
<td>1e − 3</td>
</tr>
<tr>
<td>batch size</td>
<td>64</td>
<td>max epochs</td>
<td>100</td>
</tr>
<tr>
<td>L</td>
<td>2</td>
<td>ReLU</td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>−0.1</td>
<td>#negative sample</td>
<td>2</td>
</tr>
<tr>
<td>λ₁</td>
<td>0.01</td>
<td>λ₂</td>
<td>0.2</td>
</tr>
<tr>
<td>λ₃</td>
<td>1</td>
<td>λ₄</td>
<td>1e − 5</td>
</tr>
</tbody>
</table>

E Experiment Details

E.1 Political Perspective Detection
To examine whether our learned representations of political actors would benefit perspective analysis, we replace TransE in Feng et al. (2021) with our learned representations. We use the GRGCN setting in Feng et al. (2021) as model backbone. We maintain the same evaluation settings to ensure a fair comparison and highlight the effectiveness of our learned representations compared to TransE.

E.2 Roll Call Vote Prediction
We make our best effort to maintain the same experiment settings as Mou et al. (2021) while their might be minor differences. For the random setting, we conduct roll call vote prediction for legislators in the 114th and 115th congress. We follow the same 6:2:2 split. For the time-based setting, we use the 114th congress as training and validation set and the 115th congress as test set.

E.3 Expert Knowledge Prediction
We collect expert knowledge about legislators from two political think tanks, which assign a continuous score $s$ from 0 to 1 to indicate how well a political actor aligns with their agenda. We construct a classification task from expert knowledge by creating five labels: strongly favor ($0.9 \leq s \leq 1$), favor ($0.75 \leq s \leq 0.9$), neutral ($0.25 \leq s \leq 0.75$), oppose ($0.1 \leq s \leq 0.25$), and strongly oppose ($0 \leq s \leq 0.1$). In this way, we adapt from expert knowledge to derive liberal and conservative labels for legislators and thus $D = 5$ for Equation 12. We use 7:2:1 to partition them into training, validation and test sets. We calculate evaluation metrics on the liberal and conservative set separately, and present the harmonic mean of metrics. In this way, the presented results accurately and comprehensively reflect how our approach and baselines perform on both political think tanks. For text-based baselines, we encode Wikipedia summaries of entities with these methods and predict their stances with two fully connected layers. For graph-based baselines, we train them with the constructed HIN and the expert knowledge training objective.

E.4 Figure 4
As detailed in Section E.3, each entity in PAR’s HIN has two set of scores: liberal scores $(l₀, l₁, l₂, l₃, l₄)$ and conservative scores $(c₀, c₁, c₂, c₃, c₄)$, denoting the probability that the entity is strongly against, against, neutral, favor, or strongly favor liberal/conservative values. We use $\sum_{i=0}^{4} i \cdot l_i$ and $\sum_{i=0}^{4} i \cdot c_i$ to obtain continuous values of state stances and let them be the height of blue and red bars in Figure 4.

E.5 Figure 5
We use the liberal scores of governors learned by PAR $(l₀, l₁, l₂, l₃, l₄)$ to infer their ideological stances. Specifically, we use "very conservative", "lean conservative", "lean liberal", and "very liberal" to denote that $\text{argmax}(l₀, l₁, l₂, l₃, l₄) = 0, 1, 3, 4$ respectively. In addition, there are no governor that has $\text{argmax}(l₀, l₁, l₂, l₃, l₄) = 2$ so "neutral" is omitted from the figure.