UPPAM: A Unified Pre-training Architecture for Political Actor Modeling based on Language

Xinyi Mou[1], **Zhongyu Wei**[1,2]*, **Qi Zhang**[3], **Xuanjing Huang**[3,4]
[1]School of Data Science, Fudan University, China
[2]Research Institute of Intelligent and Complex Systems, Fudan University, China
[3]School of Computer Science, Fudan University, China
[4]Shanghai Collaborative Innovation Center of Intelligent Visual Computing, China
{xymou20,zywei,qz,xjhuang}@fudan.edu.cn

Abstract

Modeling political actors is at the core of quantitative political science. Existing works have incorporated contextual information to better learn the representation of political actors for specific tasks through graph models. However, they are limited to the structure and objective of training settings and can not be generalized to all politicians and other tasks. In this paper, we propose a **U**nified **P**re-training Architecture for **P**olitical **A**ctor **M**odeling based on language (**UPPAM**). In UPPAM, we aggregate statements to represent political actors and learn the mapping from languages to representation, instead of learning the representation of particular persons. We further design structure-aware contrastive learning and behavior-driven contrastive learning tasks, to inject multidimensional information in the political context into the mapping. In this framework, we can profile political actors from different aspects and solve various downstream tasks. Experimental results demonstrate the effectiveness and capability of generalization of our method.

1 Introduction

Political actors are shaping our attitudes, opinions, and decisions toward public issues. For instance, on social platforms, politicians can select and emphasize certain aspects of content to bias the discussion, through which they can derive an opinion climate from user engagement and acquire direct feedback from potential voters and opinion leaders (Bene, 2017; Heiss et al., 2019). Political actor modeling is essential for quantitative political science and has applications in various downstream tasks such as roll call vote prediction (Yang et al., 2020), frame detection (Johnson et al., 2017) and bias detection (Baly et al., 2020).

Data-driven approaches utilize different kinds of information to profile political actors, including public statements, legislative behaviors and social networks (Figure 1). Early research analyzes roll call data to estimate the ideology of political actors. Ideal point model (Clinton et al., 2004) is one of the most widely used approaches for vote-based analysis that reveals how cleavages between legislators reflect partisan affiliation. Researchers further incorporate texts of bills to enhance the ideal point model (Gerrish and Blei, 2011, 2012; Kraft et al., 2016) and develop multidimensional vectors to replace one-dimension points. Recently, more abundant information has been considered to learn effective representations for political actors, such as co-sponsorship network (Yang et al., 2020), relations of contributors (Davoodi et al., 2020), stakeholders (Davoodi et al., 2022), mention in documents (Pujari and Goldwasser, 2021), and expert knowledge (Feng et al., 2021, 2022).

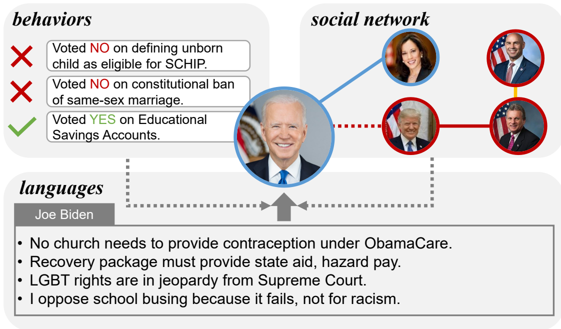


Figure 1: An illustration of political actors. They not only participate in legislative activities, but also form relationships with others, and convey opinions through tweets, speeches and etc. We propose to represent political actors based on their statements and learn the mapping from language to their representations using social networks and behaviors as self-constructed supervision.

Generally speaking, previous research aims to learn representations for a certain group of political actors using supervision from specific downstream tasks as objectives. Although they report positive results on target tasks, their models lack generalization ability in two aspects. (1) Representations

*Corresponding author.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 11996–12012

are learned on labeled data from specific tasks, e.g., state-level vote prediction, therefore they cannot be easily transferred to other tasks or scenarios. (2) The model is limited to the training setting and can not be adapted to dynamic social contexts. In other words, it's hard for the model to estimate new legislators, non-voting candidates and other political actors unseen.

Recently, large-scale pre-trained language models (PLMs) (Devlin et al., 2019; Liu et al., 2019b; Brown et al., 2020) have demonstrated a strong generalization ability and achieved excellent performance in many language modeling tasks. Motivated by PLMs, we explore representing political actors based on their statements and propose a **U**nified **P**re-training Architecture for **P**olitical **A**ctor **M**odeling based on language (**UPPAM**) [1]. We employ a two-stage training procedure following the fashion of PLMs. Firstly, we pre-train our model to learn the mapping from statements to actor representation. We propose a multigranular method to represent political actors based on language, and information of political scenarios is further injected into our model via proposed structure-aware contrastive learning and behavior-driven contrastive learning tasks. Secondly, we fine-tune the model for downstream tasks using the corresponding supervised objectives.

UPPAM is novel in three points. (1) We learn the mapping from statements to the representation of political actors, instead of directly learning actor representations. By doing so, the mapping parameters can be transferred to any downstream tasks easily, learning representations for unseen political actors based on their statements. (2) We propose several self-training tasks to inject general knowledge in the political scenarios into mapping parameters in the pre-training stage. (3) We propose a multigranular actor representation model, that can capture nuances of both general ideology and specific preferences between different political actors. We evaluate our approach on three types of tasks in quantitative political science, i.e., profile of actors, prediction of behaviors and analysis of languages. UPPAM outperforms general PLMs and other political domain-specific PLMs on these tasks. Our task-agnostic model also achieved competitive results compared to the task-specific models that employ architectures crafted for the vote prediction task. Further analysis shows the effectiveness and robustness of UPPAM in few-shot settings and different aggregation settings.

[1]We have made our code publicly available at https://github.com/xymou/UPPAM.

2 Political Actors Modeling based on Language

2.1 Multigranular Actor Representation

Political actors manifest themselves in political activities in multiple granularities. On the one hand, they hold a general ideology or bias, which is long-term and stable. On the other hand, when discussing or taking action on different issues, they hold specific positions (Gerrish and Blei, 2012), which are the result of long-term bias and short-time interests (Spell et al., 2020). Based on this, we propose to represent political actors in two granularities to model both broad ideology and specific preferences for various downstream scenarios.

General and Specific Statements Collection In practice, we use all statements a political actor has posted to get his **general** representation, characterizing the broad political leaning. Furthermore, issue-related content is adopted to help capture specific attitudes. Concretely, we use a handcrafted information retriever (see more details in Appendix A.2), to collect statements related to the queried policy area as input to encode the **specific** representation.

Statements Aggregator Since a political actor can post thousands of statements, the first challenge is how to aggregate one's statements to get his representation. It is too expensive in time and computation cost to combine full sentences. Instead, we identify indicator words from statements for information aggregation. According to the framing theory (Entman, 1993), entities and subjective content an author uses can implicitly reflect his political leaning. Following this, we identify entities, frame and sentiment words as indicators. We sort them by TFIDF (Jones, 1972) scores and keep indicators with the highest values to form an indicator sequence. In this case, for each political actor, we can get two kinds of indicator sequences, given a query about policy area j:

$$S_i^g = w_1^g, w_2^g, ...w_N^g \tag{1}$$

$$S_i^{p_j} = w_1^{p_j}, w_2^{p_j}, ...w_M^{p_j} \tag{2}$$

where S_i^g is calculated from all the statements made by political actor i, $S_i^{p_j}$ is calculated from content

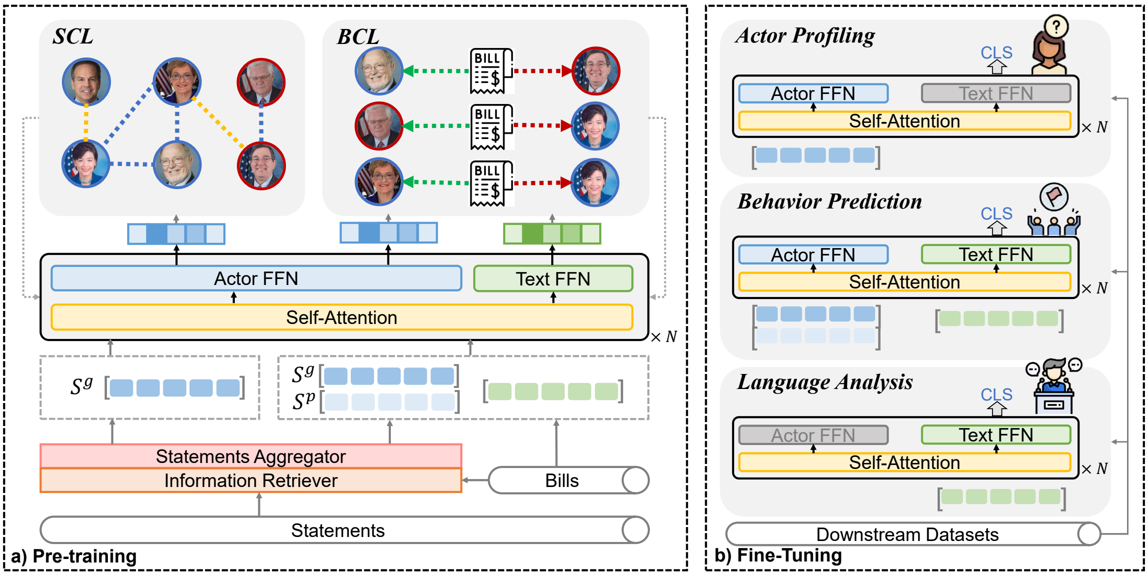


Figure 2: The proposed framework architecture. a) Pre-training. Structural and behavioral information is injected into the parameters of PLMs through the pre-training tasks. b) Fine-tuning. The same pre-trained model parameters are used to initialize models for different downstream tasks. Different modules are activated according to the downstream task type.

related to policy area j, and we reserve top N and M indicators with highest TFIDF value, where N and M are pre-defined hyper-parameters.

In subsequent pre-training and downstream tasks, we use general sequences as input when the goal is to profile the characters broadly, e.g., estimating ideology. And we input both sequences and average the representation when specific attitudes are required in tasks, as shown in Figure 2. Note that even if the issue-related content can not be retrieved, we can use the general sequence as a substitute, to ensure input compatibility.

2.2 Multidimensional Pre-training for Political Actor Modeling

To inject general knowledge of the political landscape into the mapping from statements to representation, we construct self-supervised tasks based on structural and behavioral information.

2.2.1 Structure-aware Contrastive Learning (SCL)

In terms of structural information, we mainly focus on the relationship formed between political actors. Previous studies have revealed that homophily exists in political communities, where people with similar ideologies form a link with each other (Barberá, 2015). We use two parts of links, namely party affiliation and co-sponsorship in voting. We treat party affiliation as a coarse relationship and co-sponsorship as a fine relationship respectively. By doing this, the model can further capture nuances across parties as well as inside the same party.

Party Affiliation Link We compare statements of legislators from different parties. We choose a legislator as the *anchor*, and then take another legislator with the same party affiliation as the *positive* sample, while those from the opposite party are regarded as *negative* samples. By comparing general statement sequences of legislators from different parties, the model can learn the differences in the languages of different ideologies.

Co-sponsorship Link In the legislative process, a bill is initialized by a sponsor and several co-sponsors. We assume that the more two legislators collaborate, the more they are alike since they reach agreements on many occasions (Yang et al., 2020; Mou et al., 2021). Given an *anchor* legislator, other legislators are divided into three categories based on the number of times they co-sponsored with the anchor legislator: G_1 (the co-sponsorship times are above the average); G_2 (the co-sponsorship times are below the average); G_3 (they have never co-sponsored). And we further sample *positive* and *negative* samples with the rule of $G_1 < G_2 < G_3$.

Based on the triplets constructed in the above two ways, the structure-aware contrastive objective is formulated as follows:

$$\mathcal{L}_{\text{SCL}} = \sum_{t \in \mathcal{T}_{\text{SCL}}} \left[\left\| \mathbf{t}^{(\mathbf{a})} - \mathbf{t}^{(\mathbf{p})} \right\|_2 - \left\| \mathbf{t}^{(\mathbf{a})} - \mathbf{t}^{(\mathbf{n})} \right\|_2 + \delta_{\text{SCL}} \right]_+ \tag{3}$$

where $\mathcal{T}_{\text{SCL}}$ is the set of legislator triplets, $\mathbf{t}^{(\mathbf{a})}$, $\mathbf{t}^{(\mathbf{p})}$ and $\mathbf{t}^{(\mathbf{n})}$ are actor representation encoded by general sequences of anchor, positive and negative sample in triplet t, δ_{SCL} is a hyperparameter and $[\cdot]_+$ is $max(\cdot, 0)$.

Notably, this task endows the model to capture general ideology of speakers from their languages.

2.2.2 Behavior-driven Contrastive Learning (BCL)

When it comes to behavioral information, we pay attention to the most common and important actions, i.e., voting. Specifically, we sample triplets consisting of an *anchor* bill and a pair of legislators, where the *positive* legislator $\mathbf{p}$ votes *yea* on the given bill and the *negative* one $\mathbf{n}$ votes *nay*. Different from the ideology cleavages modeled in Sec 2.2.1, the divergence of specific preferences is supposed to be reflected in the languages here. Thus, for each legislator, we extract statements about the policy area of the anchor bill as the specific sequence, input with the general sequence, as we mentioned in Sec 2.1. In this way, the behavior-driven contrastive objective is as follows:

$$\mathcal{L}_{\text{BCL}} = \sum_{t \in \mathcal{T}_{\text{BCL}}} \left[\left\| \mathbf{t}^{(\mathbf{a})} - \mathbf{t}^{(\mathbf{p})} \right\|_2 - \left\| \mathbf{t}^{(\mathbf{a})} - \mathbf{t}^{(\mathbf{n})} \right\|_2 + \delta_{\text{BCL}} \right]_+ \tag{4}$$

where $\mathcal{T}_{\text{BCL}}$ contains all vote triplets, and δ_{BCL} is a hyperparameter. $\mathbf{t}^{(\mathbf{a})}$ is the bill representation, $\mathbf{t}^{(\mathbf{p})}$ and $\mathbf{t}^{(\mathbf{n})}$ are the average of representation of the general sequence and the specific sequence, for the positive and negative legislators respectively.

It's noticeable that this pattern is not limited to the roll-call vote scenarios, instead, it can be applied to model the preferences towards any bills, events, or targets with a text description.

3 Pre-training Process

3.1 Language Model Co-training

As mentioned in Sec 2.2.2, modeling political actors in political scenarios inevitably requires encoding textual information of the bills and issues they interact with, e.g., Equation 4. Meanwhile, it is important to understand their opinions in a single discourse without context. Thus, we incorporate additional modules to model political texts. Specifically, as shown in Figure 2, we have two FFN layers in parallel in each transformer layer, to handle text and actor sequences separately. Given a sequence of input $x = \{x_1, ..., x_n\}$, the model first performs multi-head self-attention and then the corresponding module FNN_k obtains the required representation:

$$\boldsymbol{h}_k = \text{FNN}_k \left(\text{Self-Attention } (\{x_1, \ldots, x_n\}) \right) \tag{5}$$

where $k \in \{0, 1\}$ indicates the modules of actor and text respectively.

We adopt a masked language model objective to pre-train the language model. As mentioned before, political bias and framing effect are often reflected in the selection and mention of specific entities, subjective content, and emphasized frames. Thus, we take a masking strategy that upsamples entity tokens, sentiment words (Wilson et al., 2005) and frame indicators (Roy and Goldwasser, 2020) to be masked for the MLM objectives, with a 30% probability. More details can be found in Appendix B.

3.2 Overall Pre-training

Since the indicator sequence is not a normal sentence, we don't train the MLM task with contrastive learning together. Instead, the pre-training process is divided into two stages. In the first stage, we adopt the MLM task on the original statement sentences and activate text modules, to urge the model to understand the political text. Then, based on this checkpoint, we further conduct the multidimensional pre-training for political actor modeling by combining the objectives:

$$\mathcal{L}_{\text{CL}} = \alpha * \mathcal{L}_{\text{SCL}} + (1 - \alpha) * \mathcal{L}_{\text{BCL}} \tag{6}$$

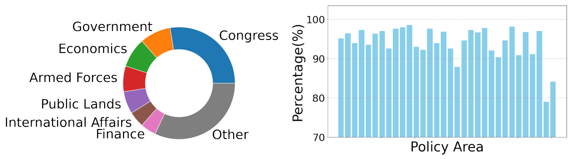
where α is hyperparameters.

4 Experiment Setup

We fine-tune our model on different kinds of downstream tasks in quantitative political science. We then compare it with prior general PLMs and political domain-specific PLMs.

4.1 Pre-training Datasets

Compared to other political actors, congress legislators are more typical and they generate massive content every day. Thus, we start with legislators to construct our pre-training datasets.



(a) (b)

Figure 3: (a) The proportion of bills in different policy areas; (b) The percentage of legislators whose related tweets can be retrieved for each policy area;

4.1.1 Public Statements of Legislators

We obtained the Twitter accounts of members of Congress from Mou et al.. On the basis of it, we further crawl data of legislators elected after 2020 and tweets of all legislators up to April 2022. Overall, we get 887 legislators and delete the meaningless tweets including self-promotion advertisements, notifications, etc., using regular expressions. Finally, the cleaned data contains 2,020,938 tweets, covering discussions of events in various areas. We keep 10K held-out tweets as the validation set.

4.1.2 Legislative Context

We collect the party affiliation, sponsorship lists of bills, bills, and corresponding voting records from *VoteView*[2] and the website of U.S. Congress[3]. Each bill belongs to a specific policy area and has textual information of title and description. We get bills of 112th and 113th for pre-training and reserve those of 114th and 115th for the formulation of downstream tasks. In the pre-training stage, 1,045 bills and 375,440 voting records are involved.

To correlate legislators' votes to their statements in the related policy area, we filtered each legislator's tweets in each policy area by the handcrafted information retriever mentioned in Sec 2.1. We finally acquire 1,142,587 tweets, and the details can be found in Appendix A.2. The distribution of the policy agenda of bills and the percentage of legislators whose related tweets can be retrieved in each policy area are shown in Figure 3a and Figure 3b. Over 90% of legislators can be retrieved with relevant statements in most policy areas.

4.2 Implementation Details

UPPAM is produced via continued pre-training on RoBERTa-base model (Liu et al., 2019b), where we add parallel FFN modules in each transformer layer with the same initialization as the original one. In the first stage, the model is trained on tweets, to minimize the MLM loss with AdamW (Loshchilov and Hutter, 2018) optimizer. In the second stage, the model is further trained on indicator sequences and bill texts, to minimize the $\mathcal{L}_{CL}$. We evaluate the model every 200 training steps on the validation set and keep the best checkpoint. The pre-training procedure takes around 96 hours on 4 Tesla V100-SXM2 GPUs. More details and hyperparameters can be found in Appendix B.

[2] `https://voteview.com/`
[3] `https://www.congress.gov/`

4.3 Downstream Tasks and Datasets

We evaluate the models on three types of tasks, namely actor profiling, behavior prediction and language analysis. Notably, datasets include not only congress legislators but also other political actors such as journalists, news media, and even anonymous users, to validate the model's generalization capability.

4.3.1 Actor Profiling

This type of task can be formulated as a user-level classification task, where we aggregate multiple statements to predict the speaker's attribute.

Ideology Detection is the main task to profile actors broadly, aiming to predict political leaning. Models are evaluated on the following datasets.

- `CongS` (Gentzkow et al., 2018) collects speeches from US congressional records.
- `celeb` (Wojcieszak et al., 2022) contains tweets of celebrities (journalists, politicians and media). We convert the ideology scores into labels according to the signs.
- `Reddit` (Kitchener et al., 2022) collects comments of common users in non-political subreddits, and labels the users with ideology in the economic dimension.
- `PEM` (Xiao et al., 2022) collects tweets of legislators, news outlets and cabinet of President Obama and President Trump.
- `TIMME` (Xiao et al., 2020) includes Twitter accounts with location information and self-identified political-polarity labels. These accounts are not run by politicians.

4.3.2 Behavior Prediction

This type of task can be regarded as a relation prediction task, where we predict a political actor's attitude or action towards a given target with a piece of text description.

Vote Prediction tasks aim to predict votes of legislators towards bills with stances of yea or nay. We follow two configurations in (Mou et al., 2021).

Method	ID					VP		GP		SD			FD		
	CongS	celeb	Reddit	PEM	TIMME	VoteIn	VoteOut	NRA	LCV	poldeb	election	SEval	twitter	gvfc	immi
BERT	81.19	69.72	62.86	87.52	84.92	84.95	83.54	49.14	65.99	61.14	72.49	65.93	49.93	76.98	62.96
RoBERTa	85.74	70.54	65.75	86.36	84.83	87.35	84.61	50.18	67.29	64.34	76.76	69.57	52.37	**81.03**	65.04
SSciBERT	82.77	70.78	61.33	81.78	83.73	85.99	84.01	49.66	64.03	59.65	69.40	64.28	50.49	76.16	61.83
POLITICS	84.73	70.67	**68.22**	90.51	84.92	86.88	84.58	48.57	66.68	63.74	73.98	71.06	50.89	78.23	62.60
PoliBERTweet	80.68	70.24	61.69	82.36	85.61	87.32	84.77	48.43	65.67	62.42	**80.12**	70.07	52.43	76.15	61.80
UPPAM	**86.82**	**71.97**	64.31	**92.09**	**85.87**	**90.30**	**86.07**	**51.54**	**69.17**	**65.24**	76.43	**71.94**	**53.99**	80.93	**67.59**

Table 1: Macro F1 scores on different evaluation tasks (average of 3 runs, more details can be found in Appendix C.3). ID, VP, GP, SD, and FD are short for Ideology Detection, Vote Prediction, Grade Prediction, Stance Detection, and Frame Detection, respectively.

- `VoteIn` refers to the in-session setup, where we randomly split the bills in the same congress session, i.e., the 114th session.
- `VoteOut` refers to the more challenging out-of-session setup, where we use data in the 114th session for training and validation while testing on the 115th session.

Grade Prediction tasks are designed as classification tasks for ratings in a certain issue, given a politician's statements and background description of the given issue. We include datasets as follows:

- `NRA` Grades (Pujari and Goldwasser, 2021) provides politicians' grades {A, B, C, D & F} assigned by *National Rifle Association* and their statements on *guns*, as well as background information of *guns* from `ontheissues.org`.
- `LCV` Grades (Pujari and Goldwasser, 2021) is similar to NRA Grades, but it's about the scores in the *environment* area.

4.3.3 Language Analysis

In addition to the overall characterization of political actors, we also test models' ability to understand individual discourses. We apply stance detection and frame detection as downstream tasks, which can be formulated as sentence-level classification tasks.

Stance detection tasks aim to predict one's stance towards a given target. The tasks take a 3-way label (favor, against, and neutral) or binary label (favor, against). We test on these datasets.

- `poldeb` (Somasundaran and Wiebe, 2010) provides opinion–target pairs from several debating platforms covering different domains.
- `election` (Kawintiranon and Singh, 2021) contains tweets related to the 2020 US presidential election, expressing stances towards President Trump and Biden.
- `SEval` (Mohammad et al., 2016) is a shared task to detect stances in public tweets.

Frame detection tasks aim to detect which frame dimensions are employed in a piece of text. It's a multi-label classification task with a pre-defined label set. We test on these datasets.

- `twitter` (Johnson et al., 2017) annotates tweets of politicians with 17 general frames.
- `gvfc` (Liu et al., 2019a) collects news headlines about gun violence, and annotates them with 9 issue-specific frame dimensions.
- `immi` (Mendelsohn et al., 2021) collects immigration-related tweets posted by the public, annotated with 14 general frames.

5 Experiment Results

5.1 Main Results

The compared general PLMs include *BERT* (Devlin et al., 2019) and *RoBERTa* (Liu et al., 2019b). We also compare our model with available PLMs for social science texts-*SsciBERT* (Shen et al., 2022), and for the political domain: *POLITICS* (Liu et al., 2022) and *PoliBERTweet* (Kawintiranon and Singh, 2022). We fine-tune all the PLMs in the same settings, and we select the best fine-tuned model on validation sets using macro F1. The implementation details and hyperparameters can be found in Appendix C.2. Table 1 presents macro F1 scores on the downstream tasks.

Actor Profiling Our model shows superior performance on various political actor modeling tasks. Results of ideology detection tasks indicate that our model can not only characterize the ideology of legislators but is also good at modeling other roles like journalists in the `celeb` dataset and cabinet in the `PEM` dataset, demonstrating the transferability of using languages to represent characters. The reason for not performing best on the `Reddit` dataset

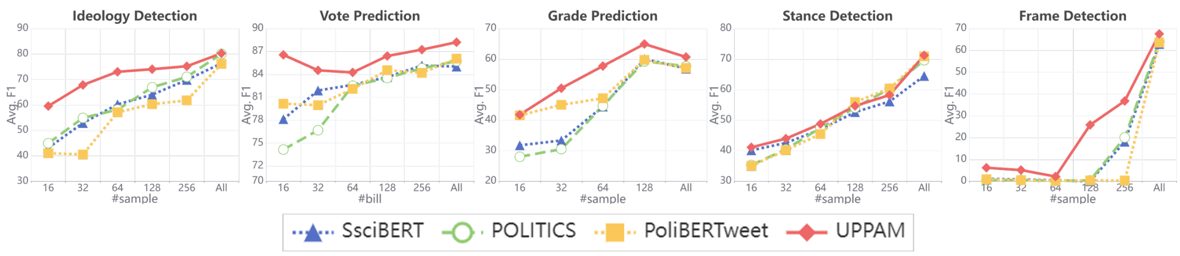


Figure 4: Average F1 on different tasks in the few-shot learning experiments. Note that for vote prediction tasks, we use # of bills and corresponding voting records, instead of # of records.

Method	ID	VP	GP	SD	FD
UPPAM	80.21	88.19	60.36	71.20	67.50
w/o SCL	78.73	87.46	60.17	68.84	65.92
w/o BCL	79.00	87.78	58.48	69.69	65.41
w/o text modules	79.01	86.95	62.05	69.38	62.93

Table 2: Results of ablation studies. Average F1 scores of different tasks are reported.

may be the gap between the expression habits of common users and that of politicians. Nevertheless, we still outperform the majority of baselines.

Behavior Prediction All the models show excellent performance on vote prediction and grade prediction tasks, using languages to represent political actors. It indicates that it's a feasible scheme to infer political actors' behaviors from their languages. Among all the PLMs, our model is the best. We attribute the performance gain to our proposed behavior-driven pre-training task.

Language Analysis Moreover, our model also achieves competitive performance on tasks of analyzing individual text including stance detection and frame detection, indicating that the ability to understand political languages is preserved while the model is learning to profile actors, benefiting from the co-training process in Sec 3.1.

5.2 Ablation Study

To explore the effects of different components, we conduct ablation studies and results are reported in Table 2. Removing SCL or BCL mainly hurts the performance of actor profiling tasks. Removing the text modules results in the most loss in language analysis tasks, especially the frame detection task. This demonstrates the necessity of separate modules to guarantee the ability to model political text.

5.3 Further Analysis

Few-shot Learning We fine-tune PLMs on different numbers of samples. Figure 4 shows UPPAM outperforms the baselines on nearly all the tasks. Benefiting from the pre-training stages, our model can better capture ideology and preference differences, even when using only 16 samples.

Method	VoteIn	VoteOut
CNN+meta (Kornilova et al., 2018)	83.40	75.89
LSTM+GCN (Yang et al., 2020)	85.85	80.59
Vote (Mou et al., 2021)	88.36	82.32
Vote+MTL (Mou et al., 2021)	88.72	83.73
UPPAM	**90.30**	**86.07**

Table 3: Comparison with the previous state-of-art models on vote prediction task. The results are reported in macro F1, on bills of the 114th and 115th congress.

Compare with Task-specific Models Taking the vote prediction task as an example, we compare our model with previous task-specific models, where particular meta-data and structural information is crafted for the task. Table 3 shows that UPPAM achieves competitive results, indicating that we can deduce political actors' votes from languages. Additionally, our method can be used to analyze non-voting actors, relieving the cold-start problem.

Methods of Statements Aggregation We show the impact of statements aggregation methods on ideology detection in fine-tuning. We mainly compare our method with *concat* (Table 4) and *mean pooling* (Table 5). *concat* means to concatenate each speaker's political statements into a flat sequence and then encode it. *mean pooling* encodes each sentence individually and uses the averaged representation as the final representation. We further discuss the impact of the number of aggregated sentences in Appendix C.2.2. Results illustrate that our model shows robustness in several settings and our aggregator is more effective and efficient.

5.4 Visualization

General Ideology We perform Principle Component Analysis (PCA) on political actor representation generated by our model for the CongS dataset.

Method	CongS	celeb	Reddit	PEM	TIMME
BERT	56.31	64.25	60.19	81.00	78.90
RoBERTa	58.99	66.65	**64.04**	77.73	77.83
SSciBERT	60.87	64.57	56.80	77.86	73.11
POLITICS	63.73	67.88	63.13	79.17	82.45
PoliBERTweet	54.48	59.54	61.37	63.35	83.46
UPPAM	**66.89**	**70.59**	61.90	**82.92**	**84.97**

Table 4: Macro F1 scores on ideology detection tasks where statements are aggregated by concatenation in fine-tuning.

Method	CongS	celeb	Reddit	PEM	TIMME
BERT	80.91	68.27	62.74	83.57	82.79
RoBERTa	86.49	70.78	63.39	85.85	84.84
SSciBERT	81.66	68.36	62.25	83.03	82.86
POLITICS	**87.43**	70.89	61.38	86.41	84.92
PoliBERTweet	78.88	67.98	63.11	85.84	86.31
UPPAM	84.71	**71.12**	**63.91**	**86.98**	**87.07**

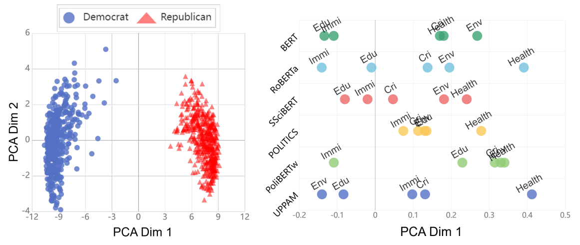
Table 5: Macro F1 scores (average of 3 runs) on ideology detection tasks where 32 statements are mean pooled in fine-tuning.

As shown in Figure 5a, our method can well separate politicians of different ideologies.

Individual Specific Preferences We also visualize specific representation in different policy areas for individuals. Figure 5b shows the representation in several highly-discussed policy areas, learned by different models from the tweets of Rep. Rooney. We can observe that Rep. Rooney behaves conservatively in *immigration*, but expresses left-wing views on *environment* (Pujari and Goldwasser, 2021). While most of our baselines fail to capture this nuance, UPPAM can well compare the relative polarity in each area.

6 Related Work

Political Actor Modeling focuses on modeling attributes and behaviors of political actors, with special attention to estimating the ideology. Because of the publicity and typicality, politicians like legislators have been the research subject for most work. The most widely used approach to estimate the ideology of legislators is ideal point model (Clinton et al., 2004) that represents legislators and bills as points in a one-dimension latent space from the roll-call data. After that, researchers further incorporate texts of bills (Gerrish and Blei, 2011; Gu et al., 2014) to enhance the model, solving the problem of prediction on new bills. Some embedding methods are also proposed to promote learning of legislators (Kraft et al., 2016; Kornilova et al., 2018).



(a) (b)

Figure 5: (a) PCA visualization of general representation of politicians in the CongS dataset; (b) Specific representation of Rep. Rooney in policy areas.

More recently, external information including cosponsorship (Yang et al., 2020), donors (Davoodi et al., 2020), relevant stakeholders (Davoodi et al., 2022) and expert knowledge (Feng et al., 2021, 2022) is used to better learn legislator representation. They follow a mixed structure of textual encoder and graph encoder, to explicitly combine textual and structural information. Despite outstanding performance on target tasks, these methods are limited to certain settings or data, behaving inefficient in dynamic political scenarios. Thus they are hard to be transferred to all actors. By contrast, methods relying on texts (Vafa et al., 2020) provide more possibility for generalization.

Domain-specific Pre-training Based on continued pre-training on domain-specific data, domain-specific Pre-trained Language Models have shown superiority on many NLP tasks. Domain-specific PLMs have been investigated in many areas including medical (Zhang et al., 2021) and financial (Araci, 2019) domains. However, little work has explored PLMs in the political domain. Li and Goldwasser pre-trained a hierarchical LSTM for political perspective identification. Kawintiranon and Singh followed BERTweet (Nguyen et al., 2020) to train a PoliBERTweet for stance detection in elections. Liu et al. recently proposed story-level contrastive learning for ideology understanding. These researches pave the way for pre-training in the political domain, but they currently only consider training objectives at the text level and are not yet able to deal with more complex problems in this domain. Thus, our work is novel in dealing with multiple levels of practical problems.

7 Conclusion

In this paper, we propose to learn political actors from languages and inject multidimensional domain knowledge into the PLMs through structure-aware contrastive learning and behavior-driven contrastive learning. Experimental results validate the effectiveness and generalization capability of our approach.

Limitations

Our work is the first step towards unified pre-training for political actor modeling and it is limited in two aspects. In terms of data, we focus on the typical political actors, i.e., the congress legislators, and their statements, without using a larger corpus like political news. But our method can be easily scaled to a larger corpus, where we can aggregate articles of different media and consider their structure information like page links for pre-training. In terms of method, in order to improve the retrieval efficiency in both pre-training and fine-tuning, we use simple methods rather than dynamic selection methods based on embeddings to query and aggregate statements, leaving much room for future exploration.

Ethics Statement

Data Collection and Privacy Our data collection is in compliance with Twitter's terms of service and matches previous publications. Although tweets are public, when releasing data, we will share user id or tweet id rather than raw data, to minimize the privacy risk.

Political Leaning Since political identity is becoming increasingly important in American society, the models could come up with some risks if a user is mislabeled with an error affiliation, e.g., a user may be socially ostracized for their supposed political beliefs (Alkiek et al., 2022). However, the research subject in this paper is public political actors, which have been studied in political science for decades, rather than the common public. Instead, understanding the bias and behaviors of these characters can help our public avoid being polarized by their certain strategies, mitigating the potential risk.

Acknowledgements

This work is partially supported by National Natural Science Foundation of China (No. 71991471) and Science and Technology Commission of Shanghai Municipality Grant (No.21QA1400600). We would also like to thank the anonymous reviewers for their constructive feedback.

References

Kenan Alkiek, Bohan Zhang, and David Jurgens. 2022. Classification without (proper) representation: Political heterogeneity in social media and its implications for classification and behavioral analysis. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 504–522, Dublin, Ireland. Association for Computational Linguistics.

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *CoRR*, abs/1908.10063.

Ramy Baly, Giovanni Da San Martino, James Glass, and Preslav Nakov. 2020. We can detect your bias: Predicting the political ideology of news articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4982–4991.

Pablo Barberá. 2015. Birds of the same feather tweet together: Bayesian ideal point estimation using twitter data. *Political analysis*, 23(1):76–91.

Pablo Barberá, Andreu Casas, Jonathan Nagler, Patrick J Egan, Richard Bonneau, John T Jost, and Joshua A Tucker. 2019. Who leads? who follows? measuring issue attention and agenda setting by legislators and the mass public using social media data. *American Political Science Review*, 113(4):883–901.

Marton Bene. 2017. Go viral on the facebook! interactions between candidates and followers on facebook during the hungarian general election campaign of 2014. *Information, Communication & Society*, 20(4):513–529.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Joshua Clinton, Simon Jackman, and Douglas Rivers. 2004. The statistical analysis of roll call data. *American Political Science Review*, 98(2):355–370.

Maryam Davoodi, Eric Waltenburg, and Dan Goldwasser. 2020. Understanding the language of political agreement and disagreement in legislative texts. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5358–5368, Online. Association for Computational Linguistics.

Maryam Davoodi, Eric Waltenburg, and Dan Goldwasser. 2022. Modeling U.S. state-level policies by extracting winners and losers from legislative texts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 270–284, Dublin, Ireland. Association for Computational Linguistics.

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

Robert M Entman. 1993. Framing: Towards clarification of a fractured paradigm. *McQuail's reader in mass communication theory*, 390:397.

Shangbin Feng, Minnan Luo, Zilong Chen, Qingyao Li, Xiaojun Chang, and Qinghua Zheng. 2021. Knowledge graph augmented political perspective detection in news media. *arXiv preprint arXiv:2108.03861*.

Shangbin Feng, Zhaoxuan Tan, Zilong Chen, Ningnan Wang, Peisheng Yu, Qinghua Zheng, Xiaojun Chang, and Minnan Luo. 2022. Par: Political actor representation learning with social context and expert knowledge. *arXiv preprint arXiv:2210.08362*.

Matthew Gentzkow, Jesse M Shapiro, and Matt Taddy. 2018. Congressional record for the 43rd-114th congresses: Parsed speeches and phrase counts. In *URL: https://data. stanford. edu/congress text*.

Sean Gerrish and David Blei. 2012. How they vote: Issue-adjusted models of legislative behavior. *Advances in neural information processing systems*, 25.

Sean M Gerrish and David M Blei. 2011. Predicting legislative roll calls from text. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*.

Yupeng Gu, Yizhou Sun, Ning Jiang, Bingyu Wang, and Ting Chen. 2014. Topic-factorized ideal point estimation model for legislative voting network. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 183–192.

Raffael Heiss, Desiree Schmuck, and Jörg Matthes. 2019. What drives interaction in political actors' facebook posts? profile and content predictors of user engagement and political actors' reactions. *Information, communication & society*, 22(10):1497–1513.

Kristen Johnson, Di Jin, and Dan Goldwasser. 2017. Leveraging behavioral and social information for weakly supervised collective classification of political discourse on Twitter. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 741–752, Vancouver, Canada. Association for Computational Linguistics.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

Kornraphop Kawintiranon and Lisa Singh. 2021. Knowledge enhanced masked language model for stance detection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4725–4735, Online. Association for Computational Linguistics.

Kornraphop Kawintiranon and Lisa Singh. 2022. PoliBERTweet: A pre-trained language model for analyzing political content on Twitter. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7360–7367, Marseille, France. European Language Resources Association.

Michael Kitchener, Nandini Anantharama, Simon D. Angus, and Paul A. Raschky. 2022. Predicting political ideology from digital footprints.

Anastassia Kornilova, Daniel Argyle, and Vladimir Eidelman. 2018. Party matters: Enhancing legislative embeddings with author attributes for vote prediction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 510–515, Melbourne, Australia. Association for Computational Linguistics.

Peter Kraft, Hirsh Jain, and Alexander M. Rush. 2016. An embedding model for predicting roll-call votes. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2066–2070, Austin, Texas. Association for Computational Linguistics.

Chang Li and Dan Goldwasser. 2021. Using social and linguistic information to adapt pretrained representations for political perspective identification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4569–4579, Online. Association for Computational Linguistics.

Siyi Liu, Lei Guo, Kate Mays, Margrit Betke, and Derry Tanti Wijaya. 2019a. Detecting frames in news headlines and its application to analyzing news framing trends surrounding us gun violence. In *Proceedings of the 23rd conference on computational natural language learning (CoNLL)*.

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

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*, pages 1354–1374, Seattle, United States. Association for Computational Linguistics.

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

Julia Mendelsohn, Ceren Budak, and David Jurgens. 2021. Modeling framing in immigration discourse on social media. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2219–2263.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. SemEval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41, San Diego, California. Association for Computational Linguistics.

Xinyi Mou, Zhongyu Wei, Lei Chen, Shangyi Ning, Yancheng He, Changjian Jiang, and Xuanjing Huang. 2021. Align voting behavior with public statements for legislator representation learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1236–1246, Online. Association for Computational Linguistics.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

Rajkumar Pujari and Dan Goldwasser. 2021. Understanding politics via contextualized discourse processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1353–1367, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shamik Roy and Dan Goldwasser. 2020. Weakly supervised learning of nuanced frames for analyzing polarization in news media. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7698–7716, Online. Association for Computational Linguistics.

Si Shen, Jiangfeng Liu, Litao Lin, Ying Huang, Lin Zhang, Chang Liu, Yutong Feng, and Dongbo Wang. 2022. Sscibert: A pre-trained language model for social science texts. *arXiv preprint arXiv:2206.04510*.

Swapna Somasundaran and Janyce Wiebe. 2010. Recognizing stances in ideological on-line debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 116–124, Los Angeles, CA. Association for Computational Linguistics.

Gregory Spell, Brian Guay, Sunshine Hillygus, and Lawrence Carin. 2020. An Embedding Model for Estimating Legislative Preferences from the Frequency and Sentiment of Tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 627–641, Online. Association for Computational Linguistics.

Keyon Vafa, Suresh Naidu, and David Blei. 2020. Text-based ideal points. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5345–5357.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 347–354, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Magdalena Wojcieszak, Andreu Casas, Xudong Yu, Jonathan Nagler, and Joshua A Tucker. 2022. Most users do not follow political elites on twitter; those who do show overwhelming preferences for ideological congruity. *Science advances*, 8(39):eabn9418.

Zhiping Xiao, Weiping Song, Haoyan Xu, Zhicheng Ren, and Yizhou Sun. 2020. Timme: Twitter ideology-detection via multi-task multi-relational embedding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2258–2268.

Zhiping Xiao, Jeffrey Zhu, Yining Wang, Pei Zhou, Wen Hong Lam, Mason A Porter, and Yizhou Sun. 2022. Detecting political biases of named entities and hashtags on twitter. *arXiv preprint arXiv:2209.08110*.

Yuqiao Yang, Xiaoqiang Lin, Geng Lin, Zengfeng Huang, Changjian Jiang, and Zhongyu Wei. 2020. Joint representation learning of legislator and legislation for roll call prediction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 1424–1430. International Joint Conferences on Artificial Intelligence Organization. Main track.

Taolin Zhang, Zerui Cai, Chengyu Wang, Minghui Qiu, Bite Yang, and Xiaofeng He. 2021. SMedBERT: A knowledge-enhanced pre-trained language model with structured semantics for medical text mining. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5882–5893, Online. Association for Computational Linguistics.

A Data Cleaning and Retrieval

In this section, we provide the details of data cleaning and how we implement a query mechanism to obtain relevant statements from the corpus.

A.1 Data Cleaning Steps

Remove Meaningless Tweets We have observed that some tweets of the legislators do not express opinions and are unrelated to this research, such as the self-promotion advertisements, and notifications. Thus, we delete these tweets using regular expressions. Some examples of filter patterns are shown in Table 6.

Type	Filter Patterns
self-promotion	video release, don't miss, watch live, watch here, I'll be on live
notifications	deadline for,breaking:
personal life	my family, my daughter, my son, my husband, my wife

Table 6: Examples of patterns used to filter out meaningless tweets.

Clean Tweets We replace urls and user mentions with symbols `[URL]` and `[MENTION]`.

A.2 Information Retriever

In order to improve the efficiency of the retrieval, we do not use dynamic methods such as sentence embedding similarity but based on previous work (Barberá et al., 2019; Pujari and Goldwasser, 2021) to implement manual rules for querying.

Firstly, we get all policy areas of US politics and their description and codebook from https://www.congress.gov/ and https://www.comparativeagendas.net/. We then extract nouns and adjectives and delete stopwords from the description of each policy area, to form the keywords for each policy area. Since the words used in the definition can be abstract such as *economy*, we further summarize hashtags for specific issues following (Pujari and Goldwasser, 2021). The process is divided into 3 steps. (1) We counted hashtags in the corpus and retained hashtags that at least 100 members used. (2) Then, we merge hashtags for the same event or issue based on co-occurrence. (3) At last, we mapped events or issues to policy areas through the wiki of events and the codebook of policy areas to form a mapping from hashtags to policy areas. Overall, we use the keywords and hashtags to retrieve tweets of each member. Table 7 shows some examples.

Policy Area	keywords	hashtags
Civil Rights	discrimination, race, gender, disability, equal, abortion,	#rosaparks, #reprorights, #righttochoose, #hobbylobby, ...
Health	treatment, disease, health, medicare, medicaid, drug	#lowerdrugcosts, #covid19, #aca, #trumpcare ...
Immigration	immigration, refugee, immigrant, smuggling,	#daca, #dreamact, #homeishere, #refugee ...

Table 7: Examples of keywords and hashtags used to retrieve tweets related to given policy areas.

B Pre-training Details

This section illustrates some details in the continued pre-training part.

Training Details. UPPAM is produced via continued pre-training on RoBERTa-base model (Liu et al., 2019b), where we add parallel FFN modules in each transformer layer with the same initialization as the original one. In this way, our model contains about 153M parameters. Our implementation is based on the HuggingFace Transformers library[4]. The hyperparameters are listed in Table 8.

MLM Strategy We link entities using DBPedia spotlight[5] with types of *person*, *organization* and *event*. We identify sentiment words and frame indicators using lexicons by (Wilson et al., 2005) and (Roy and Goldwasser, 2020). We mask these tokens with a 30% probability, and randomly mask the remaining tokens with a 15% probability. As done in (Devlin et al., 2019), the masked tokens are replaced with `[MASK]`, random tokens and the original tokens with a ratio of 8:1:1.

Construction of Triplets When generating triplets using the co-sponsorship information, we may get member triplets with a pattern of "<D, R, D>" or "<R, D, R>", where D and R represent Democrat and Republican. These samples can contradict some samples generated according to party affiliation. Thus, we deleted samples in these formats.

C Fine-tuning Details

C.1 Fine-tuning Datasets

This section lists more details of the datasets used in our downstream evaluation. Statistics are listed in Table 9.

[4] https://github.com/huggingface/transformers
[5] https://www.dbpedia-spotlight.org/

Hyperparameter	Value
number of steps	9,600 for stage1; 2,890 for stage 2
batch size	2048
maximum learning rate	2e-5
learning rate scheduler	linear decay with warmup
warmup percentage	10%
optimizer	AdamW
δ_{SCL}	1
δ_{BCL}	1
α	0.5
N	256
M	256

Table 8: Hyperparameters used in continued pre-training.

Data	# Train actors	# Train records
CongS (Gentzkow et al., 2018)	861	344,478
celeb (Wojcieszak et al., 2022)	1,690	715,643
Reddit (Kitchener et al., 2022)	1,865	178,115
PEM (Xiao et al., 2022)	407	825,179
TIMME (Xiao et al., 2020)	1,808	974,732
VoteIn (Mou et al., 2021)	506	129,869
VoteOut (Mou et al., 2021)	506	149,122
NRA (Pujari and Goldwasser, 2021)	206	4,377
LCV (Pujari and Goldwasser, 2021)	219	5,725
poldeb (Somasundaran and Wiebe, 2010)	-	4,993
election (Kawintiranon and Singh, 2021)	-	1,575
SEval (Mohammad et al., 2016)	-	2,251
twitter (Johnson et al., 2017)	-	1,420
gvfc (Liu et al., 2019a)	-	910
immi (Mendelsohn et al., 2021)	-	1,627

Table 9: Statistics of downstream datasets.

- `CongS` (Gentzkow et al., 2018): We use the speaker's party affiliation as the ideology label, following (Liu et al., 2022).

- `celeb` (Wojcieszak et al., 2022): We convert the ideology scores into ideology labels where those negative are converted to left-leaning labels while those positive are converted into right-leaning labels. We crawl the tweets posted after 01/01/2020 of these celebrities. We assume their ideologies do not change during the period.

- `Reddit` (Kitchener et al., 2022): The original paper collected 91,000 reddit users. For the time being, we have succeeded to crawl 3,918 users and their comments.

- `PEM` (Xiao et al., 2022) includes accounts of legislators in the 115th and 116th congresses, well-known news outlets, Obama, Trump and their cabinet members. We include 582 accounts they publicly provided.

- `TIMME` (Xiao et al., 2022) includes 2,584 Twitter accounts with location information and self-identified political-polarity labels (either Democratic or Republican).

- `VoteIn` & `VoteOut` (Mou et al., 2021): For the in-session setup, we randomly select 20% bills for testing, 10% is for validation and the rest for training. For the out-of-session settings, we train and validate on bills of the 114th congress and test on that of the 115th congress. For both settings, bills in the test are unseen in training.

- `NRA` & `LCV` (Pujari and Goldwasser, 2021): we use the formatted statements and tweets provided by the paper to predict the NRA and LCV rankings, which are originally National Rifle Association (NRA) scores and League of Conservation Voters (LCV) scores.

- `poldeb` (Somasundaran and Wiebe, 2010): covers debates in domains of the existence of god, healthcare, gun rights, gay rights, abortion and creationism.

- `election` (Kawintiranon and Singh, 2021): includes tweets expressing support or opposition towards Trump or Biden during the 2020 US election period.

- `SEval` (Mohammad et al., 2016): The dataset contains stances towards six targets: Atheism, Climate Change, Feminist, Hillary Clinton, Abortion, and Donald Trump.

- `twitter` (Johnson et al., 2017): Tweets are annotated with 17 frame dimensions, covering 6 issues, i.e., abortion, aca, guns, immigration, isis and lgbt.

- `gvfc` (Liu et al., 2019a): 1,300 headlines of news articles on gun violence, annotated with 9 issue-specific frames.

- `immi` (Mendelsohn et al., 2021): We use the tweets which are annotated with 14 general frames.

C.2 Fine-tuning Procedure

C.2.1 Fine-tuning in Main Experiments

Ideology Detection. We aggregate general statements of the speakers using the method mentioned in Sec 2.1. Then we encode the sequence and use the representation of `[CLS]` token for classification. We only activate actor modules during fine-tuning.

Hyperparameter	Value
# epochs	20
batch size	16
patience of early stopping	5
maximum learning rate	1e-5 or 2e-5
maximum sequence length	256
optimizer	AdamW
weight decay	1e-4
# FFN layer	1
hidden layer dimension	768
dropout	0.5

Table 10: Hyperparameters used in fine-tuning. maximum sequence length is 128 for PoliBERTweet.

Vote Prediction & Grade Prediction. We encode the bills or issues using text modules. And we aggregate both general statements and specific statements about the bill's policy area or given issue to represent legislators using the actor modules. Then we calculate the dot product of the representation and apply an FFN for classification.
Stance Detection. We formulate a simple input by concatenating the target and the text and use the [CLS] token for standard fine-tuning. We only activate text modules during fine-tuning.
Frame Detection. We use the [CLS] token for standard fine-tuning of sequence classification (Devlin et al., 2019). The threshold for multilabel classification is set to 0.5 for all models. We only activate text modules during fine-tuning.

Fine-tuning hyperparameters are listed in Table 10.

C.2.2 Fine-tuning Experiments of Different Aggregation Methods

Concatenation Limited by the length of the input sequence of PLMs, we select political-related content using the method in Appendix A.2. And we will truncate the concatenated sequence if it has more than 512 tokens.
Mean Pooling Due to computational resource constraints, instead of encoding all statements of a person, we randomly sample records of his political-related statements as input. And the averaged sentence embeddings are used as actor representation.
Impact of the Number of Aggregated Statements We further explore the impact of the number of statements when using mean pooling to acquire political actor representation in the fine-tuning process. Figure 6 illustrates the average macro F1 of different models on ideology detection tasks.

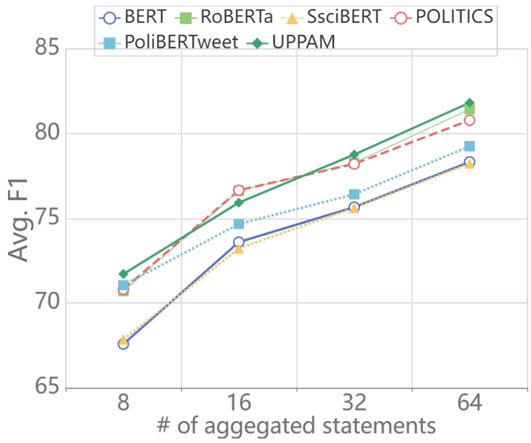


Figure 6: Average macro F1 on ideology detection tasks when aggregating different numbers of statements by mean pooling. We didn't include more sentences due to the constraint of computation resources.

Method	ID				
	CongS	celeb	Reddit	PEM	TIMME
BERT	$81.19_{\pm 1.63}$	$69.72_{\pm 1.76}$	$62.86_{\pm 2.21}$	$87.52_{\pm 3.73}$	$84.92_{\pm 1.07}$
RoBERTa	$85.74_{\pm 0.97}$	$70.54_{\pm 1.94}$	$65.75_{\pm 1.90}$	$86.36_{\pm 3.00}$	$84.83_{\pm 0.43}$
SSciBERT	$82.77_{\pm 4.11}$	$70.78_{\pm 2.49}$	$61.33_{\pm 4.04}$	$81.78_{\pm 4.46}$	$83.73_{\pm 2.05}$
POLITICS	$84.73_{\pm 1.81}$	$70.67_{\pm 1.18}$	$68.22_{\pm 2.74}$	$90.51_{\pm 1.71}$	$84.92_{\pm 0.48}$
PoliBERTweet	$80.68_{\pm 5.14}$	$70.24_{\pm 4.63}$	$61.69_{\pm 4.44}$	$82.36_{\pm 2.00}$	$85.61_{\pm 2.38}$
UPPAM	$86.82_{\pm 0.80}$	$71.97_{\pm 1.79}$	$64.31_{\pm 1.90}$	$92.09_{\pm 0.97}$	$85.87_{\pm 1.44}$

Table 11: Average macro F1 and standard deviations on ideology detection tasks.

Method	VP		GP	
	VoteIn	VoteOut	NRA	LCV
BERT	$84.95_{\pm 1.14}$	$83.54_{\pm 0.06}$	$49.14_{\pm 12.02}$	$65.99_{\pm 14.74}$
RoBERTa	$87.35_{\pm 0.06}$	$84.61_{\pm 0.42}$	$50.18_{\pm 12.62}$	$67.29_{\pm 18.56}$
SSciBERT	$85.99_{\pm 0.58}$	$84.01_{\pm 0.21}$	$49.66_{\pm 13.34}$	$64.03_{\pm 14.91}$
POLITICS	$86.88_{\pm 0.73}$	$84.58_{\pm 0.17}$	$48.57_{\pm 14.89}$	$66.68_{\pm 17.11}$
PoliBERTweet	$87.32_{\pm 0.74}$	$84.77_{\pm 0.32}$	$48.43_{\pm 14.14}$	$65.67_{\pm 16.13}$
UPPAM	$90.30_{\pm 0.22}$	$86.07_{\pm 0.18}$	$51.54_{\pm 12.65}$	$69.17_{\pm 14.08}$

Table 12: Average macro F1 and standard deviations on vote prediction and grade prediction tasks. Large stand deviation on GP tasks is the result of the change of test set. Since the sizes of NRA and LCV datasets are quite small, we change the test set for different runs, while keeping the test set always the same for other tasks.

Method	SD		
	poldeb	election	SEval
BERT	$61.14_{\pm 1.21}$	$72.49_{\pm 1.94}$	$65.93_{\pm 0.84}$
RoBERTa	$64.34_{\pm 1.19}$	$76.76_{\pm 1.51}$	$69.57_{\pm 1.01}$
SSciBERT	$59.65_{\pm 1.17}$	$69.40_{\pm 0.57}$	$64.28_{\pm 1.77}$
POLITICS	$63.74_{\pm 0.84}$	$73.98_{\pm 1.31}$	$71.06_{\pm 0.91}$
PoliBERTweet	$62.42_{\pm 1.33}$	$80.12_{\pm 0.36}$	$70.07_{\pm 0.59}$
UPPAM	$65.24_{\pm 1.06}$	$76.43_{\pm 0.23}$	$71.94_{\pm 1.01}$

Table 13: Average macro F1 and standard deviations on stance detection tasks.

We can observe that utilizing more sentences can improve the performance, where aggregating 64 statements can achieve competitive results of our

Method		FD	
	twitter	gvfc	immi
BERT	49.93$\pm_{1.66}$	76.98$\pm_{0.79}$	62.96$\pm_{0.82}$
RoBERTa	52.37$\pm_{3.13}$	81.03$\pm_{2.68}$	65.04$\pm_{2.36}$
SSciBERT	50.49$\pm_{0.64}$	76.16$\pm_{1.77}$	61.83$\pm_{1.02}$
POLITICS	50.89$\pm_{1.53}$	78.23$\pm_{2.46}$	62.60$\pm_{2.38}$
PoliBERTweet	52.43$\pm_{6.15}$	76.15$\pm_{4.32}$	61.80$\pm_{1.20}$
UPPAM	53.99$\pm_{0.89}$	80.93$\pm_{1.33}$	67.59$\pm_{0.93}$

Table 14: Average macro F1 and standard deviations on frame detection tasks.

proposed indicator sequence method. However, this method costs 64 times more training time than our method.

C.3 Fine-tuning Results

Table 11, Table 12, Table 13 and Table 14 show the standard error of our 3 runs of fine-tuning on downstream tasks.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
We discuss the limitations in the "Limitation" section.

☑ A2. Did you discuss any potential risks of your work?
Please refer to the "Ethic Statement" section.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Please refer to the "Introduction" section.

☒ A4. Have you used AI writing assistants when working on this paper?
No.

B ☑ Did you use or create scientific artifacts?

Please refer to section 2&3.

☑ B1. Did you cite the creators of artifacts you used?
Please refer to section 4.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We use multiple existing open-source artifacts that are based on different licenses, making it difficult to summarize. We cite the resources of utilized artifacts where the license details can be found.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Please refer to section 4.

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Please refer to section 4 and Appendix C.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Please refer to section 4.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Please refer to section 4&5 and Appendix B&C.

C ☑ Did you run computational experiments?

Please refer to section 4&5.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Please refer to section 4 and Appendix B&C.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Please refer to section 4 and Appendix B&C.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Please refer to section 5 and Appendix B&C.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Please refer to section 4 and Appendix A.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?
Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.