Joint Representation Learning of Legislator and Legislation for Roll Call Prediction

Yuqiao Yang[1,2*] , **Xiaoqiang Lin**[1*] , **Geng Lin**[1] , **Zengfeng Huang**[1] ,
Changjian Jiang[4†] and **Zhongyu Wei**[1,3†] ,
[1]School of Data Science, Fudan University, China
[2]School of Information and Communication, Tokyo Institute of Technology, Japan
[3]Research Institute of Intelligent and Complex Systems, Fudan University, China
[4]School of International Relations & Public Affairs, Fudan University, China
yyqfaust@gmail.com, {xqlin16, glin16, huangzf, Changjian, zywei}@fudan.edu.cn

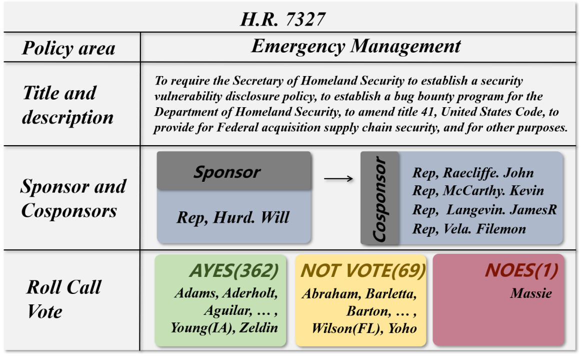
Abstract

In this paper, we explore to learn representations of legislation and legislator for the prediction of roll call results. The most popular approach for this topic is named the ideal point model that relies on historical voting information for representation learning of legislators. It largely ignores the context information of the legislative data. We, therefore, propose to incorporate context information to learn dense representations for both legislators and legislation. For legislators, we incorporate relations among them via graph convolutional neural networks (GCN) for their representation learning. For legislation, we utilize its narrative description via recurrent neural networks (RNN) for representation learning. In order to align two kinds of representations in the same vector space, we introduce a triplet loss for the joint training. Experimental results on a self-constructed dataset show the effectiveness of our model for roll call results prediction compared to some state-of-the-art baselines.

1 Introduction

Quantitative political science aims to understand how the government behaves via analyzing legislative data. Roll call data, historical records of legislators' votes on a set of issues, attracts intensive attention from quantitative political scientists because it presents actions of law-makers. Researchers have analyzed roll call data from the United States Congress [Clinton *et al.*, 2004] or the British Parliament [Cowley and Garry, 1998] to reveal the political leaning of congressmen. An example of a legislative record can be seen in Figure 1. Each record contains voting results and information about the legislation with its title and description. A legislation is initialized by a group of congressmen, consisting of a sponsor and several co-sponsors. Voting results made by legislators include *Ayes (yes)*, *Noes (no)*, and *Not Vote*.

The most popular approach for roll call data modeling is named *ideal point model* [Clinton *et al.*, 2004]. It represents legislators and legislation as points in a latent space. And the voting behavior of a legislator can be characterized as the distance between his/her position and the target legislation in the space. With this setting, researchers explore to predict the voting results of legislators. However, the positions of legislators are learned based on their historical voting results, ignoring other context information. This results in poor predicting accuracy. In practice, the voting behavior of the congressman is affected by multiple factors, the topic of legislation, the influence of other congressmen, personal preferences, etc. Although some extensions [Gerrish and Blei, 2011] are made to incorporate textual information for better modeling the legislation, the ideal point model still lacks the power to capture a richer context of the legislative record.



H.R. 7327			
Policy area	*Emergency Management*		
Title and description	*To require the Secretary of Homeland Security to establish a security vulnerability disclosure policy, to establish a bug bounty program for the Department of Homeland Security, to amend title 41, United States Code, to provide for Federal acquisition supply chain security, and for other purposes.*		
Sponsor and Cosponsors	*Sponsor* *Rep, Hurd. Will*	→	*Cosponsor* *Rep, Raecliffe. John* *Rep, McCarthy. Kevin* *Rep, Langevin. JamesR* *Rep, Vela. Filemon*
Roll Call Vote	*AYES(362)* *Adams, Aderholt, Aguilar, … , Young(IA), Zeldin*	*NOT VOTE(69)* *Abraham, Barletta, Barton, … , Wilson(FL), Yoho*	*NOES(1)* *Massie*

Figure 1: An example of a legislative record collected from U.S. Congress. The legislation is initialized by Rep. Hurd. Will with four co-sponsors.

To tackle this problem, we propose to learn representations for both legislators and legislation incorporating context information of a legislative record for better prediction of roll call results. For legislators, we introduce relationships among legislators and utilize graph convolutional networks (GCNs) for representation learning. For legislation, we encode its narrative description via recurrent neural networks (RNNs) for the representation learning. Following the ideal point model, we map representations of the legislation and legislators into the same embedding space. A triplet loss is utilized as the objective function to learn the two kinds of representations

[*] Indicates the two authors have equal contributions.

[†] Contact author

jointly. Therefore, legislators' degree of approval to a target legislation can be measured by the Euclidean distance between their representations and the legislation's representation. We collect a dataset from the Congress website in the U.S. between 1993 and 2018 for the evaluation. The major contributions of this paper can be three-fold.

- We represent a congressman using their background information and jointly learn dense representations of all congress members incorporating their network information via graph neural networks.
- We align representations of legislation and legislators in the same vector space and introduce a triplet loss for the joint training.
- We collect a dataset including legislative records of U.S. Congress. Experimental results from this self-constructed dataset show the effectiveness of our framework. To the best of our knowledge, this is the first dataset constructed for voting result prediction in the political area. We will make it public.

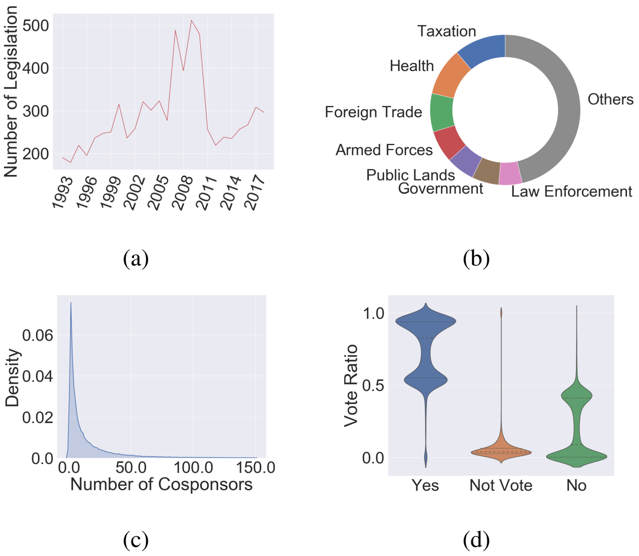
2 Dataset Construction

The dataset is collected from the Congress website in U.S. [1], including both legislation information and roll call results from 1993 to 2018. There are 215,857 legislation in total, with 2,234,082 voting records. 2,347 legislators are involved in this dataset. We present the change of numbers of legislation along the time span in Figure 2a.

Legislator. A legislator is a person who writes and passes laws, especially someone who is a member of a legislative body. Our targets here are Members of the U.S. House of Representatives. The total number of voting representatives is fixed by law at 435, and the term of office of the members is two years, with no limit for re-election (the re-election rate is around 81.91%). Every member owns an ID and has information about party and state.

Legislation. The legislation is a law which has been promulgated by a legislature or in the process of making it. Before an item of legislation becomes a law, it may be known as a bill and can be broadly referred to as "legislation" (we use legislation through this paper). Every legislation belongs to a specific policy area and it has related textual information of title and description. We list the proportion of legislation in different policy areas in Figure 2b. A legislation is initialized by a group of congressmen, consisting of a sponsor and several co-sponsors. The distribution of the numbers of co-sponsors is shown in Figure 2c.

Roll call record. A roll call record contains voting results of legislators with respect to a legislation. There are three types of voting results, namely, "yeas" (yes), "nays" (no), and "not vote". The distribution of voting results can be seen in Figure 2d.



(a)

(b)

(c)

(d)

Figure 2: (a) number of legislation each year; (b) proportion of legislation in different policy areas; (c) distribution of the number of co-sponsors for each legislation; (d) distribution of different stands in roll call results.

3 Task Formulation and Overall Architecture

Given a legislation and a set of legislators, we aim to predict the voting result for each legislator. We describe the prediction task starting with the definition of symbols.

- A set of congress members $M = \{m_1, m_2, \dots, m_k\}$ and every member m_i has an ID: $m_i(ID)$, information of party: $m_i(p)$ and state: $m_i(s)$.
- A set of legislation $L = \{l_1, l_2, \dots, l_n\}$ and every legislation l_i has a description $l_i(d)$, and a sponsor co-sponsor network $l_i(s)$.
- A set of roll call voting records $R = \{r(m_i, l_j) | 1 \leq i \leq k, 1 \leq j \leq n\}$, $r(m_i, l_j)$ stands for the voting result of legislator m_i to legislation l_j and it has three labels, namely *yea*, *nay* and *not vote*.

The overall inference workflow can be seen in Figure 3. It mainly includes three components, namely, legislation representation learning, legislator representation learning, and voting result prediction. We learn the representation of legislators with graph convolutional networks utilizing sponsor co-sponsor network and learn the representation of legislation using recurrent neural network based on its description. The voting preference of a legislator with respect to a legislation is measured by the similarity between their representations. We then rank legislators in terms of their preferences (higher preference score means higher tendency to vote *yes*) and predict their voting results according to a provided voting ratio (proportion of different voting results). The voting ratio of a legislation is automatically determined by a semantic GCN model.

For the training, we introduce a triplet loss as the objective function to jointly learn of the two kinds of representations. We will introduce our joint training framework in Section 4 and the ratio aware result prediction module in Section 5.

[1] www.congress.gov. The dataset is avaliable on http://www.sdspeople.fudan.edu.cn/zywei/data/fudan-USRollCall.zip and the codes for this paper is avaliable on https://github.com/lxqjdai/Joint-Representation-Learning-of-Legislator-and-Legislation-for-Roll-Call-Prediction

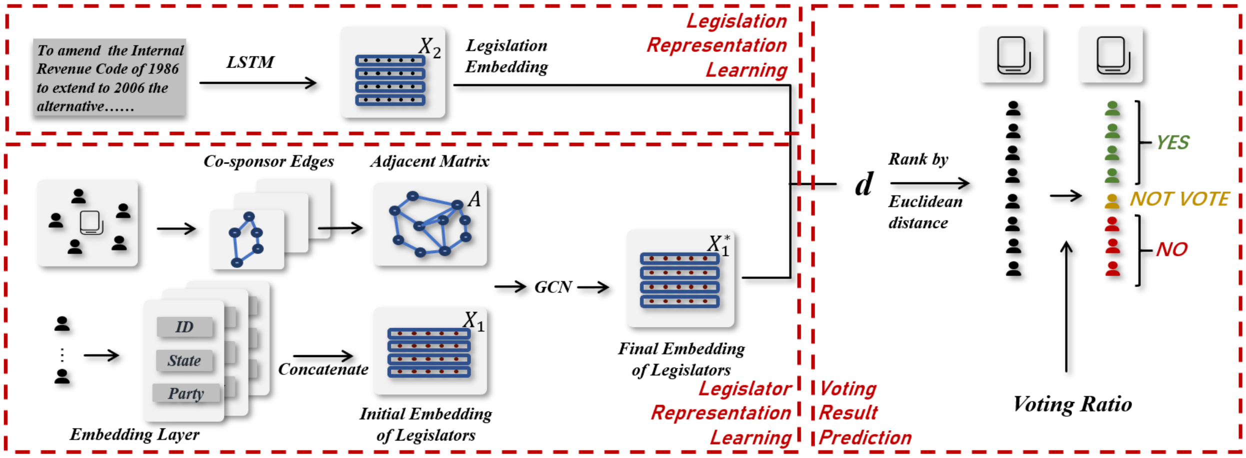

To amend the Internal Revenue Code of 1986 to extend to 2006 the alternative......
LSTM
X_2
Legislation Embedding
Legislation Representation Learning
Co-sponsor Edges
Adjacent Matrix
A
GCN
X_1^*
Final Embedding of Legislators
ID
State
Party
Concatenate
X_1
Embedding Layer
Initial Embedding of Legislators
Legislator Representation Learning
d
Rank by Euclidean distance
YES
NOT VOTE
NO
Voting Ratio
Voting Result Prediction

Figure 3: The overall structure of the inference workflow for voting result prediction.

4 Joint Representation Learning of Legislation and Legislators

We explore to learn dense vectors to include complex information for both legislation and legislators. And then, we aim to align both kinds of representations within the same vector space via joint training. The joint learning framework mainly includes three components, namely, GCN-based legislator representation learning, LSTM-based legislation representation learning, and joint training based on triplet loss.

4.1 GCN-based Legislator Representation Learning

We initialize the representation of congressmen by their background information and then update their representations via GCNs taking legislator network into consideration.

Representation initialization. We first map every congress member to a continuous low-dimension vector, X_{lgt}, with three components, namely, ID, party, and state.

$$X_{lgt}(i) = X_{ID}(i) \oplus X_{Party}(i) \oplus X_{State}(i) \qquad (1)$$

where X_{ID} stands for ID which is unique for every Congress member, X_{Party} denotes democracy or republican and X_{State} stands for the state where the member is elected.

Legislator network construction. Each legislation is initialized by a sponsor and several co-sponsors. It is reasonable that the relationship among legislators can be reflected from such political activity, i.e., raise a legislation together. In specific, we regard legislators as nodes and use the connections of sponsor and co-sponsor as the edge in the network. We consider all the sponsor and co-sponsor relationships in a period of time to obtain the adjacency matrix A. The element a_{ij} means the number of legislation that m_i and m_j co-sponsor.

GCN-based legislator representation updating. GCNs [Kipf and Welling, 2016] are neural networks operating on graphs and including features of nodes based on properties from their neighborhoods. In our scenario, we utilize GCNs for the representation updating of legislators based on the sponsor co-sponsor network. Each layer of our GCNs is structured as follows:

$$Z = f(X, A) = AReLU(AX_{input}W^{(0)})W^{(1)} \qquad (2)$$

Here, A is the normalized adjacency matrix of the network. $W^{(0)} \in \mathbf{R}^{C\times H}$ is an input-to hidden weight matrix for a hidden layer. $W^{(1)} \in \mathbf{R}^{H\times F}$ is a hidden-to-output weight matrix. Here we take X_{lgt} of all legislators as input and update their representation accordingly.

4.2 LSTM-based Legislation Representation

As for the legislation, we use the title and description for its representation learning. We use the recurrent neural network (LSTM [Hochreiter and Schmidhuber, 1997] in our case) to encode the text information of the legislation as the legislation feature(X_{lgn}).

$$X_{lgn}(j) = LSTM(l_j(d)) \qquad (3)$$

$l_i(d)$ stands for the text information of the legislation l_j.

4.3 Joint Training via Triplet Loss

For inference, the distance between legislators and legislation should represent the legislators' preference for voting for the legislation. We, therefore, want to align representations of both the legislation and legislators within the same space. In order to achieve this, we introduce triplet loss [Chechik *et al.*, 2010; Wang *et al.*, 2014] for the joint training.

At each training iteration, we sample a mini-batch of triplets. Each one consists of the legislation and a pair of legislators, one is positive ${m_j}^{+}$ and the other is negative ${m_k}^{-}$. Voting results of the two legislators should satisfy $r({m_k}^{-}, l_i) < r({m_j}^{+}, l_i)$ with the rule of $YES < Not\ Vote < No$. The goal of triplet loss is to push away the representation of negative legislator $X_{lgt}(k)^{-}$ from the representation of the legislation X_{lgn} by a distance margin $\epsilon > 0$ compared to the representation of positive legislators $X_{lgt}(j)^{+}$. Here is the form of hinge loss to implement this:

$$L = max(\epsilon, \|X_{lgn}(i) - X_{lgt}(j)^{+}\|^2 - \|X_{lgn} - X_{lgt}(k)^{-}\|^2) \qquad (4)$$

5 Ratio Aware Voting Result Prediction

Based on representations of legislators and the target legislation, we measure the preference of legislators to vote for this legislation by computing their Euclidean distances in the embedding space. Legislators are then ranked by the distance in ascending order (the closer the distance is, the higher tendency to vote for this legislation). With a provided voting ratio, i.e., the proportion of different voting results, we are able to classify legislators as supporters/neutrals/opposers in turn. In practice, we predict the ratio of voting results for the legislation automatically. We explore to utilize a GCN-based model for voting ratio prediction, named semantic-GCN.

Legislation network construction. We treat each legislation as a node and the similarity between two legislation as the weight of the edge for legislation network construction. Following [Liu *et al.*, 2019], we extract keywords for each legislation using the Text-rank algorithm and use the average of keywords' glove vectors as its representation. Based on the representation, we compute the similarity between legislation.

Semantic-GCN based voting ratio prediction. Take the representation of legislation learned in Section 4.2 as initial input, we utilize GCN to update the representation of legislation. We concatenate the GCN output with the initial representation of the legislation and predict the voting ratio through a two-layer MLP regression. Note that the output is a probability distribution with three entries corresponding to three voting results.

6 Experiment

6.1 Experiment Setup

Experimental dataset construction. Considering legislation in the dataset spread in a large range of time span, we set a window size to extract roll-call data to build a number of experimental datasets for the evaluation of our model. In specific, we continuously extract five years of data from 1993 to 2018 each time to construct a single experiment set, and this results in 22 splits with overlaps. In each split, we use data in the first four years as the training set and the data in the last year as the testing set. Moreover, we randomly select 20% of legislation in the training set for parameter tuning. For instance, the first experiment set is constructed based on legislation raised from 1993 to 1997 (training on the year 1993 to 1996 and test on the year 1997). We calculate accuracy for each experiment set and report the average accuracy of the 22 sets for evaluation.

Legislator network construction. The information of sponsor and co-sponsor network might uncover future information for roll call result prediction of legislation in the same period of time. Therefore, We use sponsor and co-sponsor information in a period of time to construct the legislator network for the representation learning of legislators. In order to avoid the issue of future information leaking, we use the legislator network constructed in the training set for the inference of legislation in the test set.

Implementation details. The GCN for legislator representation learning is two-layer and the size of the hidden layer is 32. The dimension size of the legislator representation is 32 (16 for ID, 8 for the party, 8 for state). The initial learning rate is 1×10^{-4}. We apply early stop to keep the model from over-fitting. The number of the hidden layer in the Semantic GCN is 32. The parameters are tuned on the validation set.

6.2 Models for Comparison

We compare our model with two state-of-the-art approaches.

clinton [Clinton *et al.*, 2004] This model utilizes the Bayesian model to compute positions of legislators and legislation in the ideal point space. Distance between representations of legislators and legislation is used to characterize the voting behavior.

gerrish [Gerrish and Blei, 2011] This model extends the *ideal point model* with textual information of the legislation and embeds the positions of legislation using text regression.

We employ different algorithms for three components in our framework as ablation study, namely, legislation representation learning, legislator representation learning and voting result prediction. Here are the details of comparative models.

LSTM+Deepwalk This model uses deepwalk [Perozzi *et al.*, 2014] to obtain the representation of legislators. LSTM is used for legislation representation learning. It concatenates the two representations and uses a two-layer perceptron model for voting result prediction.

LSTM+Node2vec This model is similar to *LSTM+Deepwalk* except it uses the node2vec [Grover and Leskovec, 2016] for legislator representation learning.

LSTM+GCN LSTM, GCN and a two-layer perceptron are used for legislation representation learning, legislator representation learning and voting result prediction.

LSTM+Party This model is similar to *LSTM+GCN*, except we use only party information to represent legislators.

LSTM+GCN+triplet loss This is our model that utilizes the triplet loss to update the representations of legislation and legislators jointly. The voting ratio of the legislation is obtained via semantic GCN.

LSTM+GCN+triplet loss (GT) We use the ground-truth voting ratio in our model and report the results for reference.

6.3 Overall Performance

The overall performance of different models are shown in Table 1. We have following findings:

- The performance of *clinton* is worse than all other methods, which shows the limited prediction power of the ideal point model that ignores context information of legislative records. By incorporating the textual information, *gerrish* can improve the performance by a large margin.
- By using dense representations for legislation and legislators, the prediction models achieve much better accuracy compared to the classic model *clinton*.

Model	Acc
clinton	68.10
gerrish	75.30
LSTM + Deepwalk	75.88
LSTM + Node2vec	75.89
LSTM + Party	76.26
LSTM + GCN	76.75
LSTM + GCN + triplet loss	**78.09**
LSTM + GCN + triplet loss (GT)	81.86

Table 1: Overall performance of different models for roll-call results prediction. **bold** number is the best performance.

- Compared *LSTM+GCN* with *LSTM+deepwalk* and *LSTM+node2vec*, we can see the effectiveness of using graph neural network to model relationships among legislators for their representation learning. It is interesting to note that the performance of *LSTM+party* that uses party information to stand for a legislator is quite promising. We will further analyze this in the next sub-section.
- Our model *LSTM+GCN+triplet loss* performs better than all other baselines. This indicates the effectiveness of using triplet loss for joint training the representation of legislation and legislators.
- By using the ground-truth voting ratio, the model can further improve the performance compared to models with automatic ratio prediction results.

6.4 Performance of Voting Ratio Prediction

We further analyze the performance of our semantic-GCN for automatic voting ratio prediction. We compare four different models with various configurations for legislation representation learning and adjacency matrix construction.

S-GCN-Cos-Glove We learn the initial representation of legislation via averaging the embedding of words it contains. The similarity between two legislation is computed cosines similarity of the two representation vectors. A two-layer GCN is then used to update the representation of legislation for voting ratio prediction.

S-GCN-Cos-LSTM This model is the same as *S-GCN-Cos-Glove*, except that it uses LSTM to initialize the representation of legislation.

S-GCN-Cos-Cat-LSTM On top of *S-GCN-Cos-LSTM*, this model concatenates the output of semantic GCN and LSTM as the representation of a legislation for the final prediction. This is the one we used for voting ratio prediction.

S-GCN-Coo-Cat-LSTM This model is the same as *S-GCN-Cos-Cat-LSTM*, except the adjacency matrix for GCN is built via co-occurrence of keywords.

We use MSE as the evaluation metric. The lower the score is, the better the model is. The results can be seen in Table 2. The performance of *S-GCN-Cos-LSTM* and *S-GCN-Cos-Glove* shows no significant difference. However, *S-GCN-Cos-LSTM*'s performance is relatively more stable across different years. So we choose LSTM to continue. The performance of *S-GCN-Cos-Cat-LSTM* is better than that of *S-GCN-Cos-LSTM*, which shows that the concatenation works. The performance of *S-GCN-Cos-Cat-LSTM* is apparently better than that of *S-GCN-Coo-Cat-LSTM*, which shows that the cosine similarity works better than co-occurrence when constructing the adjacency matrix.

Model	MSE
S-GCN-Cos-Glove	0.02136
S-GCN-Cos-LSTM	0.02123
S-GCN-Coo-Cat-LSTM	0.03232
S-GCN-Cos-Cat-LSTM	**0.01580**

Table 2: Performance of various configurations of semantic-GCN models for voting ratio prediction. **bold** number is the best performance.

6.5 Influence of Party Stand on Voting Behavior of Legislators

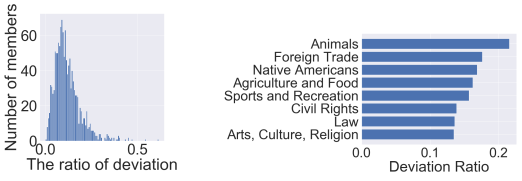
Legislators in the same party tend to have the same stand to a specific legislation. Therefore, party is a vital feature for voting result prediction. The performance of *LSTM+party* in Table 1 confirms this. In this sub-section, we look deep into the relationship between the voting behavior of legislators and their party stand.

Intra-party consistency. We first define the main stand of a party with respect to a specific legislation as the majority voting result of party members. We then measure the intra-party consistency as the percentage of legislators in the party who make the same voting decision as the main stand. Averaging on all the experimental sets for both parties, we get that the intra-party consistency is 88.42% (Democratic: 87.05% VS Republican: 89.11%). This shows that there are a number of legislators who do not follow the main stand of their party.

Deviation degree of legislator. We say that a legislator deviates from the party if he/she did not follow the main stand of the party in a voting process. We compute the deviation degree DR_{m_i} of a legislator m_i, with the following formula: $DR_{m_i} = \frac{D_{m_i}}{T_{m_i}}$. D_{m_i} is the total number of deviations of m_i, T_{m_i} is the total number of voting records of m_i. The distribution of deviation degree of legislators is shown in Figure 4a.

Behavior prediction for legislators of high deviation degree. We select the top 5% of legislators with higher deviation degree. Their behaviors are supposed to be more difficult to predict. The results of the prediction accuracy for these legislators are shown in Table 3. Experimental results show that our model is more effective for modeling behaviors of these legislators with higher deviation degree.

Deviation degree of legislation. We further study the degree of deviation for legislation to see in what policy areas legislators tend to vote differently with the main stand of their party. We define the deviation degree DR_{l_j} of legislation l_j with the formula: $DR_{l_j} = \frac{D_{l_j}}{T_{l_j}}$. D_{l_j} is the number of legislators who deviates from their party in voting this legislation and T_{l_j} is the total number of legislators involved in this voting. We then define the deviation degree of policy area P_k as the average deviation degree of legislation it includes:



(a)

(b)

Figure 4: (a) Distribution of deviation degree of legislators. We select legislators with more than 50 roll call records (accounts for 98.13%); (b) distribution of deviation degree of different policy areas (we list top 8 out of 32 to save the space)

Model	Acc
`LSTM + Deepwalk`	62.37
`LSTM + Node2vec`	62.61
`LSTM + Party`	62.85
`LSTM + GCN`	62.70
`LSTM + GCN + triplet loss`	**65.93**
`LSTM + GCN + triplet loss (GT)`	67.68

Table 3: Performance of different models for roll-call results prediction on legislators with top 5% deviation degree. **bold** number is the best performance.

$DR_{P_k} = \frac{\sum_{l_j \in P_k} D_{l_j}}{\# l_j \in P_k}$. The distribution of deviation degrees for different policy areas is shown in Figure 4b. We can infer that policy areas related to animals, food, and sports, which are highly related to the individual personalities instead of party interests, obtain a higher deviation ratio.

6.6 Case Study

To show the efficiency of our framework qualitatively, we select a specific legislation and get the representation of the legislation and legislators. Then, we project representations into the 2-D space through PCA. The visualization can be seen in Figure 5. We can see that legislators who vote for this legislation stay close to the representation of this legislation and those who against the legislation locate far away. This shows that the triplet loss efficiently encodes the roll call result information into the representation of legislators and legislation.

7 Related Work

There are two related research fields, namely, the application of Graph Neural Network and roll call result prediction.

Application of Graph Neural Network. The most popular task of Graph Neural Network explores to learn dense representation for nodes in the network. Traditional GNN models usually use explicit graph Laplacian regularization [Zhu *et al.*, 2003; Belkin *et al.*, 2006; Weston *et al.*, 2008] while current research borrows the idea from word embedding learning [Mikolov *et al.*, 2013]. Approaches in the latter category includes DeepWalk [Perozzi *et al.*, 2014], LINE [Tang *et al.*, 2015]) and node2vec [Grover and Leskovec, 2016]. Other graph embedding methods incorporate text information [Tu *et al.*, 2017; Sun *et al.*, 2016]. More recent literature, i.e., GCNs [Kipf and Welling, 2016], makes progress in combining graph structure with supervised machine learning targets directly. Based on GCNs, researchers explore different applications including semantic role labeling [Marcheggiani and Titov, 2017], named entity recognition[Cetoli *et al.*, 2017] and structured label classification[Chen *et al.*, 2017].

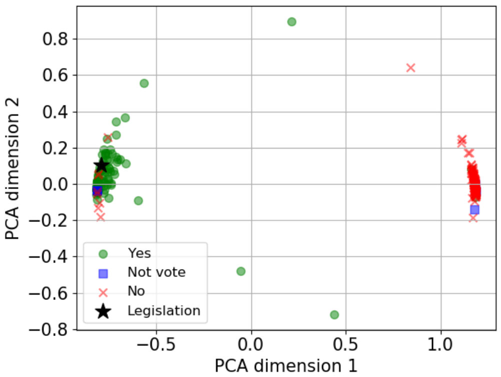


Figure 5: Layout of the representation of legislators and a specific legislation. Black star is the representation of the legislation; green circles represent those who vote ”yes” for this legislation; red crosses represent those who are against this legislation; blue squares represent those who do not vote.

Roll call result prediction. Modeling the policy preferences of legislators toward legislative outcomes is the main research problem of modern legislative behavior research. Based on legislative voting records, researchers make inferences about the policy preferences of legislators [Clinton *et al.*, 2004], the conflicted political issues [Cowley and Garry, 1998], the cohesiveness of parties [Rosenthal and Voeten, 2004] and the existence of intra-party factions [Rosenthal and Voeten, 2004]. Most of transitional research is based on ideal point model [Clinton *et al.*, 2004; Jackman and Rivers, 2004]. [Gerrish and Blei, 2012] develops the issue-adjusted ideal point model. And there are also some game-theoretic models of congressional vote prediction [Ghoshal and Honorio, 2016; Irfan and Gordon, 2018].

8 Conclusions and Future Work

We focus on the task of roll call results prediction based on legislative data. We build up the first dataset for this task and propose a framework to learn dense vectors for both the legislation and legislators to make the prediction of roll call result. Experimental results show the effectiveness of our framework. In future work, we are interested in conducting graph convolutions dynamically to help draw the picture of the change in the Congress members.

Acknowledgments

This work is partially supported by National Ministry of Science and Technology of China (No. 2018YFC0830600), National Natural Science Foundation of China (No. 61751201, No. 71991471) and Science and Technology Commission of Shanghai Municipality Grant (No. 17JC1420200, No.18DZ1201000).

References

[Belkin *et al.*, 2006] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[Cetoli *et al.*, 2017] Alberto Cetoli, Stefano Bragaglia, AD O'Harney, and Marc Sloan. Graph convolutional networks for named entity recognition. *arXiv preprint arXiv:1709.10053*, 2017.

[Chechik *et al.*, 2010] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11(Mar):1109–1135, 2010.

[Chen *et al.*, 2017] Meihao Chen, Zhuoru Lin, and Kyunghyun Cho. Graph convolutional networks for classification with a structured label space. *arXiv preprint arXiv:1710.04908*, 2017.

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

[Cowley and Garry, 1998] Philip Cowley and John Garry. The british conservative party and europe: the choosing of john major. *British Journal of Political Science*, 28(3):473–499, 1998.

[Gerrish and Blei, 2011] Sean Gerrish and David M Blei. Predicting legislative roll calls from text. In *Proceedings of the 28th international conference on machine learning (icml-11)*, pages 489–496, 2011.

[Gerrish and Blei, 2012] Sean Gerrish and David M Blei. How they vote: Issue-adjusted models of legislative behavior. In *Advances in Neural Information Processing Systems*, pages 2753–2761, 2012.

[Ghoshal and Honorio, 2016] Asish Ghoshal and Jean Honorio. From behavior to sparse graphical games: Efficient recovery of equilibria. In *Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on*, pages 1220–1227. IEEE, 2016.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Irfan and Gordon, 2018] Mohammad T Irfan and Tucker Gordon. The power of context in networks: Ideal point models with social interactions. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 910–918. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

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

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[Liu *et al.*, 2019] Bang Liu, Di Niu, Haojie Wei, Jinghong Lin, Yancheng He, Kunfeng Lai, and Yu Xu. Matching article pairs with graphical decomposition and convolutions. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 6284–6294, 2019.

[Marcheggiani and Titov, 2017] Diego Marcheggiani and Ivan Titov. Encoding sentences with graph convolutional networks for semantic role labeling. *arXiv preprint arXiv:1703.04826*, 2017.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *neural information processing systems*, pages 3111–3119, 2013.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

[Rosenthal and Voeten, 2004] Howard Rosenthal and Erik Voeten. Analyzing roll calls with perfect spatial voting: France 1946–1958. *American Journal of Political Science*, 48(3):620–632, 2004.

[Sun *et al.*, 2016] Xiaofei Sun, Jiang Guo, Xiao Ding, and Ting Liu. A general framework for content-enhanced network representation learning. 2016.

[Tang *et al.*, 2015] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. International World Wide Web Conferences Steering Committee, 2015.

[Tu *et al.*, 2017] Cunchao Tu, Han Liu, Zhiyuan Liu, and Maosong Sun. Cane: Context-aware network embedding for relation modeling. In *Meeting of the Association for Computational Linguistics*, pages 1722–1731, 2017.

[Wang *et al.*, 2014] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1393, 2014.

[Weston *et al.*, 2008] Jason Weston, Frederic Ratle, and Ronan Collobert. Deep learning via semi-supervised embedding. pages 1168–1175, 2008.

[Zhu *et al.*, 2003] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. pages 912–919, 2003.