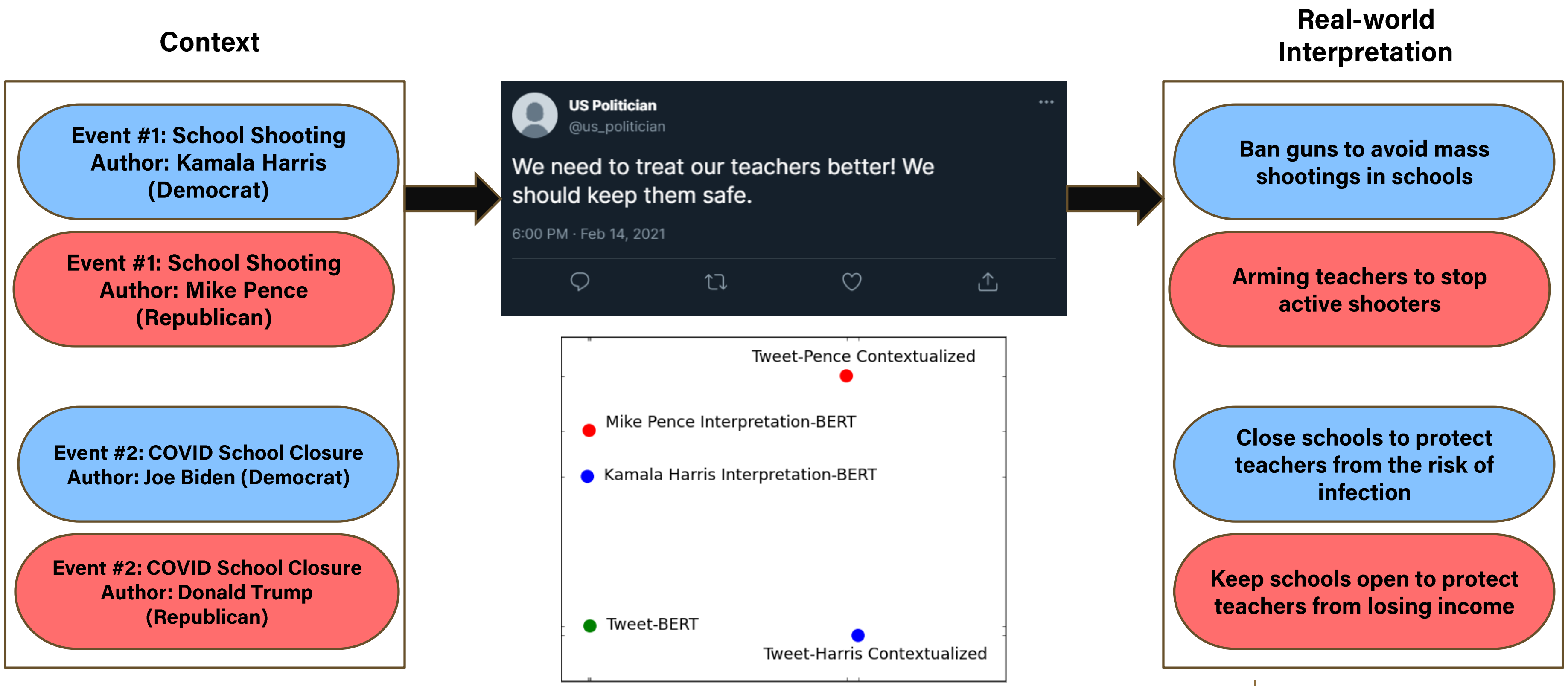
UNDERSTANDING POLITICS VIA DISCOURSE CONTEXTUALIZATION

Rajkumar Pujari and Dan Goldwasser

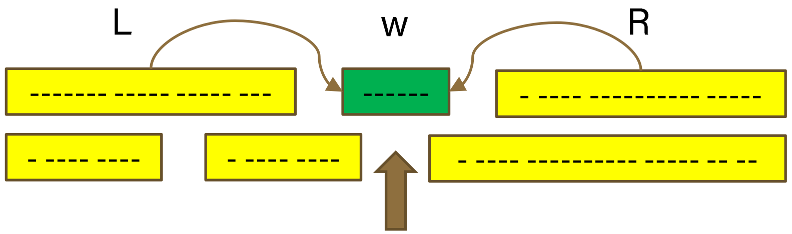
PURDUE UNIVERSITY
Department of Computer Science

MOTIVATION

- Over the last decade, political discourse has moved from traditional outlets to social media. Political text tends to be concise and subtle.



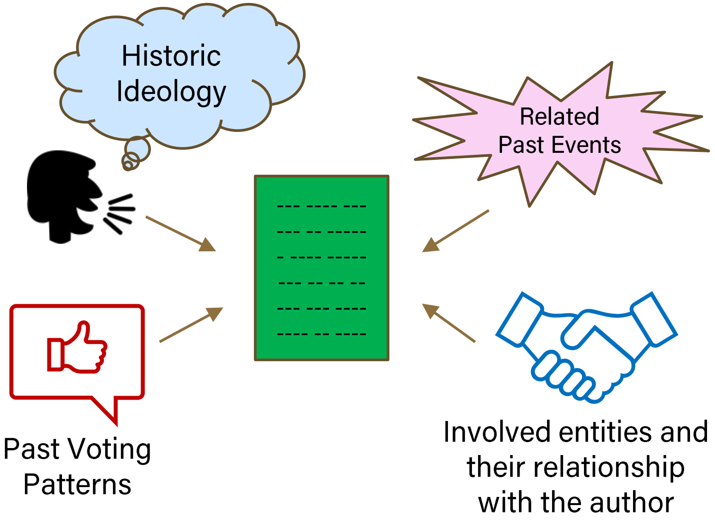
APPROACH



rep(w) ~ rep(w|L, R)

BERT, XLNET, RoBERTa etc., aim to capture left and right local textual context better in their representations

vs.



We aim to capture real-world context of the text better in our representations

Challenge: How can this massive amount of political content be used to create principled representations of politicians, their stances on issues and legislative preferences?

OVERVIEW

- Data

- Compositional Reader Pipeline
- Baseline Model
- Learning Tasks
- Evaluation Tasks
- Quantitative Evaluation Results
- Qualitative Visualization
- Conclusion

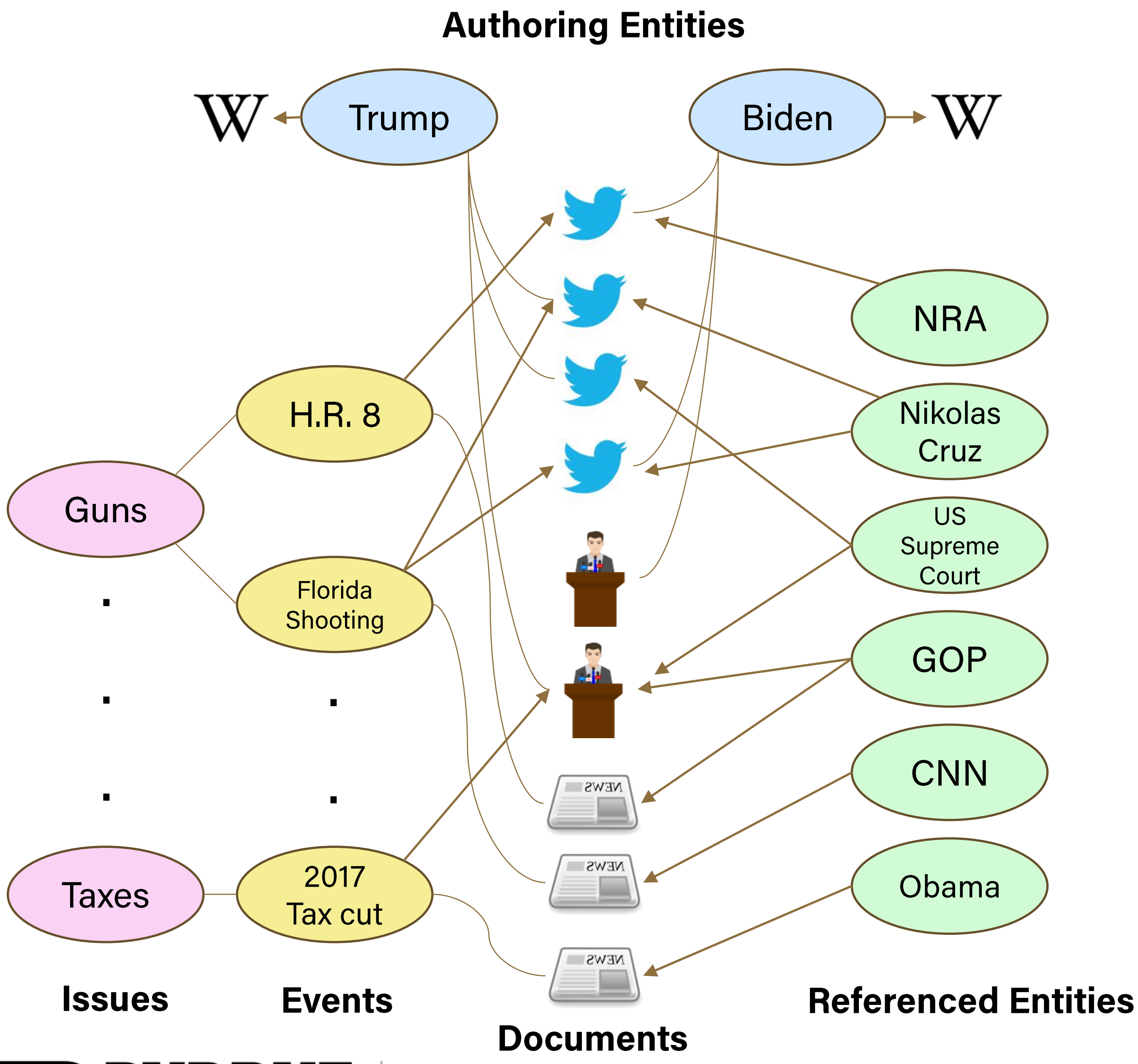
DATA SUMMARY

- We collected US political text related to **8** political issues to train and evaluate the model:
  - *Guns, LGBTQ rights, Abortion, Immigration, Economic Policy, Taxes, Middle East, and Environment*

Data	Count
News Events	367
Authoring Entities	455
Refenced Entities	10,506
Wikipedia Docs	455

Data	Count
Tweets	86,409
Press Releases	62,257
Perspectives	30,446
News Articles	8,244

An Example Text Graph



OVERVIEW

- Data
- **Compositional Reader Pipeline**
- Baseline Model
- Learning Tasks
- Evaluation Tasks
- Quantitative Evaluation Results
- Qualitative Visualization
- Conclusion

MODEL OVERVIEW

- Our model consists of three modules: Graph Generator, Encoder, and Composer
- Graph Generator takes a query triplet of <*Authors, Issues, Events*> as input. It generates a subgraph from the knowledge graph that is relevant to the input query.
- Encoder generates initial representations for each node in the Subgraph given its topology and the text of the document nodes in the subgraph
- Composer uses the graph structure to iteratively update the node embeddings

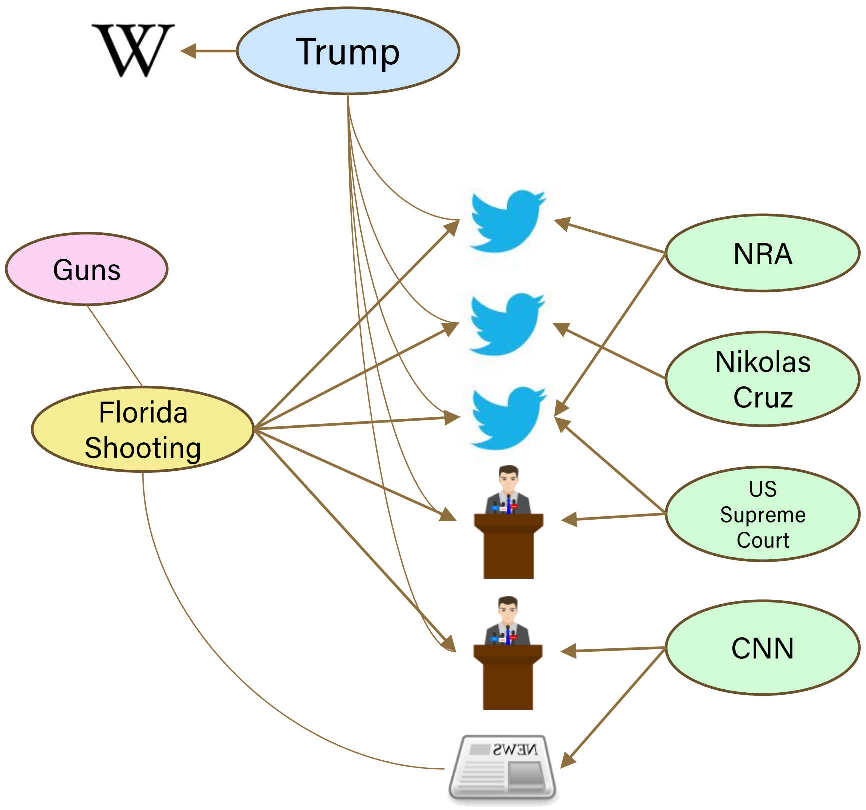
GRAPH GENERATOR

Example Query:

Authors: {Donald Trump}
Issues: {Guns}
Events: {Florida Shooting}

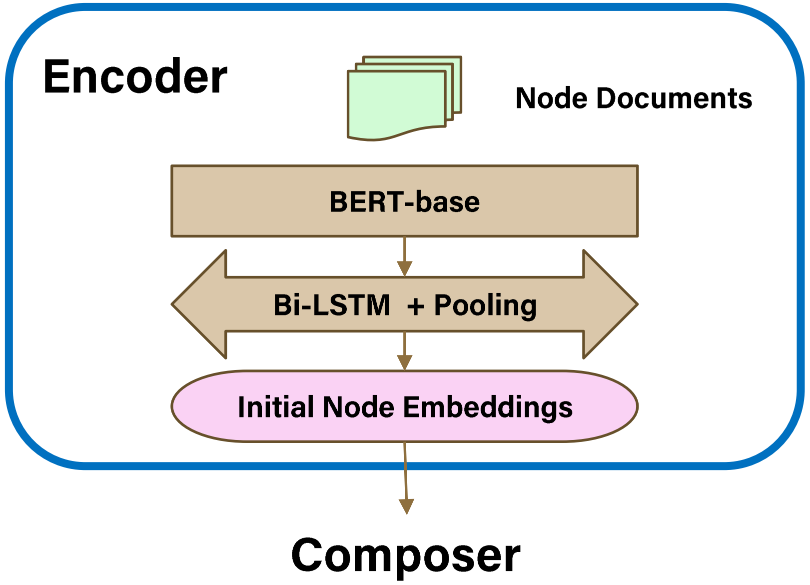
The query mechanism also allows for querying groups of authors. Hence, it could also be used to characterize consensus stances of groups such as caucuses or an entire party.

Output Subgraph:



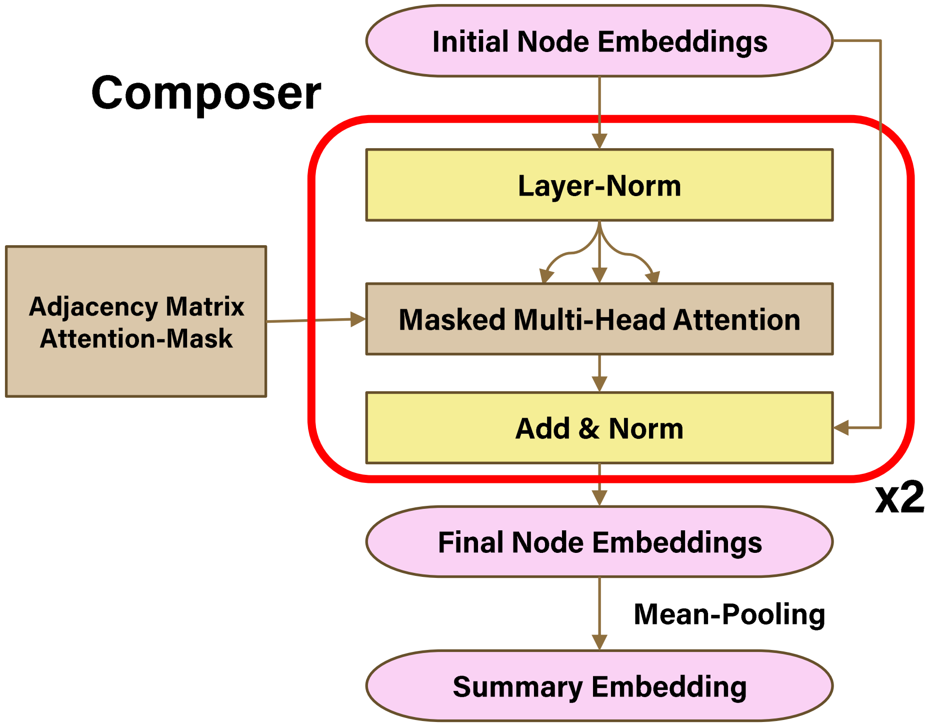
ENCODER

- Encoder is used to compute the initial node embeddings for all nodes in the subgraph. It consists of BERT (replaceable by any text representation model) followed by a Bi-LSTM.
- For a text node, the initial representation is the BERT representation of the document text.
- For each non-textual node, it takes a sequence of documents as input (documents to which a node is connected in the subgraph). The documents are ordered temporally.
- We first encode the documents using BERT and then pool them using a Bi-LSTM layer to generate initial node embeddings for non-textual nodes. The initial node embeddings are then input to the Composer module.

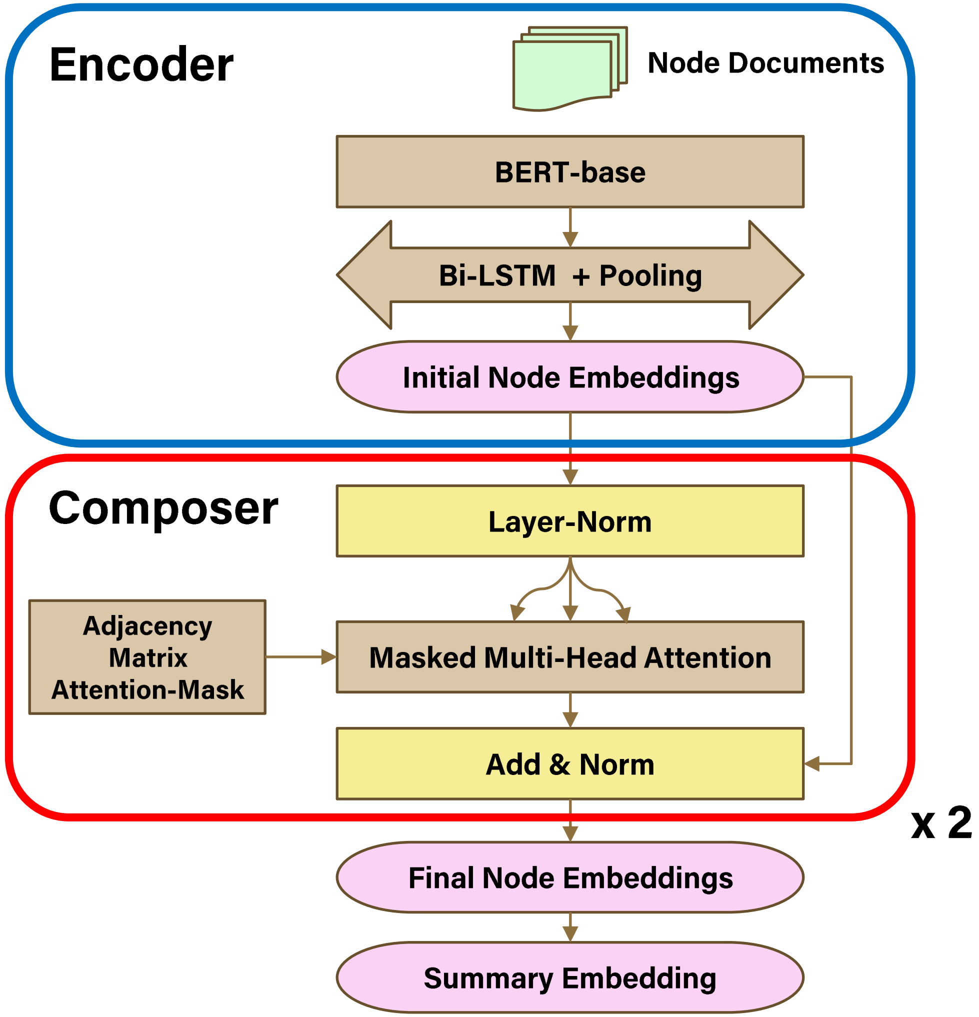


COMPOSER

- Composer module takes initial node embeddings and the adjacency matrix as inputs and generates updated node embeddings
- Composer architecture is a transformer-based Graph Attention Network (GAT) followed by a pooling layer
- We use the encoding part of the transformer architecture as the Graph Attention Network (replaceable by any graph composition network). We use the adjacency matrix of the graph as the attention-mask input.
- The final node embeddings are then mean-pooled to generate a summary embedding for the subgraph.



COMPOSITIONAL READER ARCHITECTURE



OVERVIEW

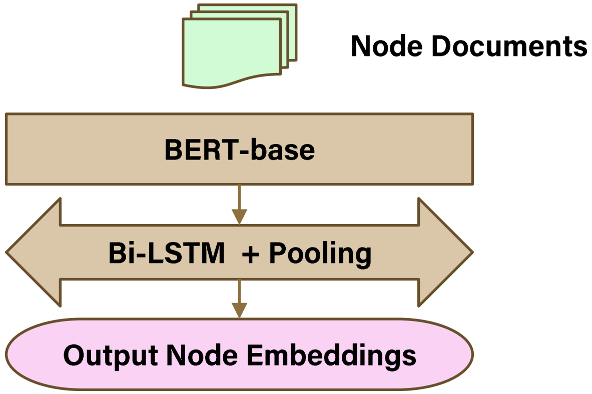
- Data
- Compositional Reader Pipeline
- **Baseline Model**
- Learning Tasks
- Evaluation Tasks
- Quantitative Evaluation Results
- Qualitative Visualization
- Conclusion

BERT ADAPTATION BASELINE

- We design BERT Adaptation baseline to benchmark our representations. Its architecture is same as the Encoder module.

- Key difference between the Adaptation baseline and Encoder output from the full-model is that the latter is "*trained on the learning tasks via back-propagation through the Composer module.*" while the baseline is trained directly on the learning tasks.

- We also compare our representations to the initial node embeddings generated by the Encoder module and pooled-BERT representation baselines.



OVERVIEW

- Data
- Compositional Reader Pipeline
- Baseline Model
- **Learning Tasks**
- Evaluation Tasks
- Quantitative Evaluation Results
- Qualitative Visualization
- Conclusion

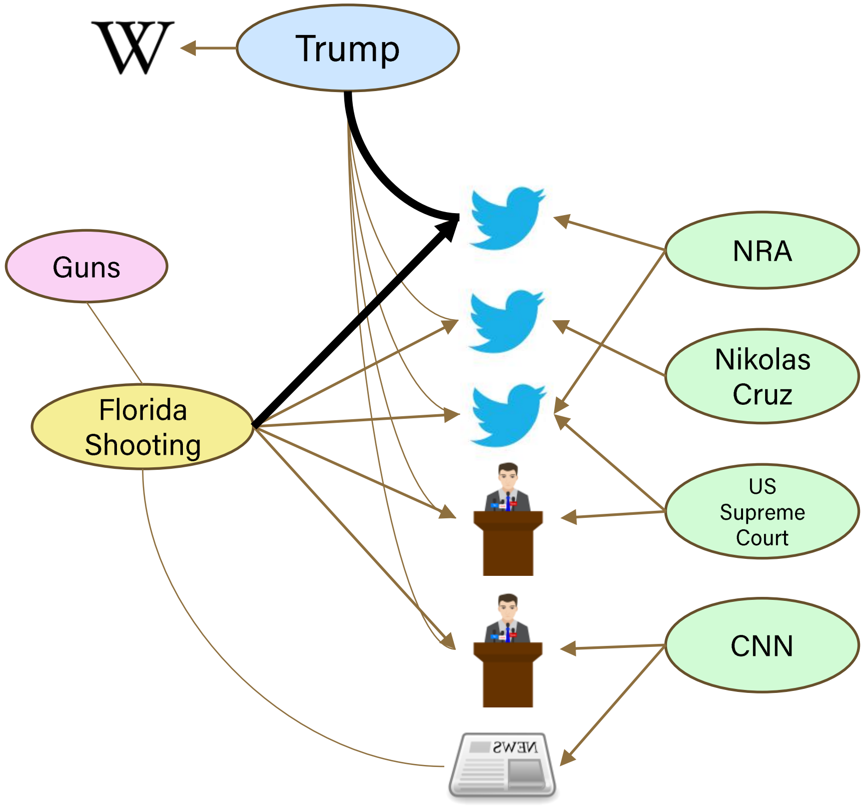
LEARNING TASKS

- We design two link prediction tasks over the generated query graphs to train the encoder-composer model:
  1) Authorship Prediction
  2) Referenced Entity Prediction

- Each task is designed with the objective of training the model to learn correlation between language in the text and the real-world context that encapsulates the text

AUTHORSHIP PREDICTION

- Intuition behind this learning task is to enable our model to learn differentiating between:
  1) language of the politician's first-person discourse vs. third person discourse of news articles
  2) language of the politician vs. language used by other politicians
  3) language of the politician in context of one issue vs. in context of other issues
- We ask the binary question: Is this tweet "***Donald Trump talking about Florida Shooting***"?

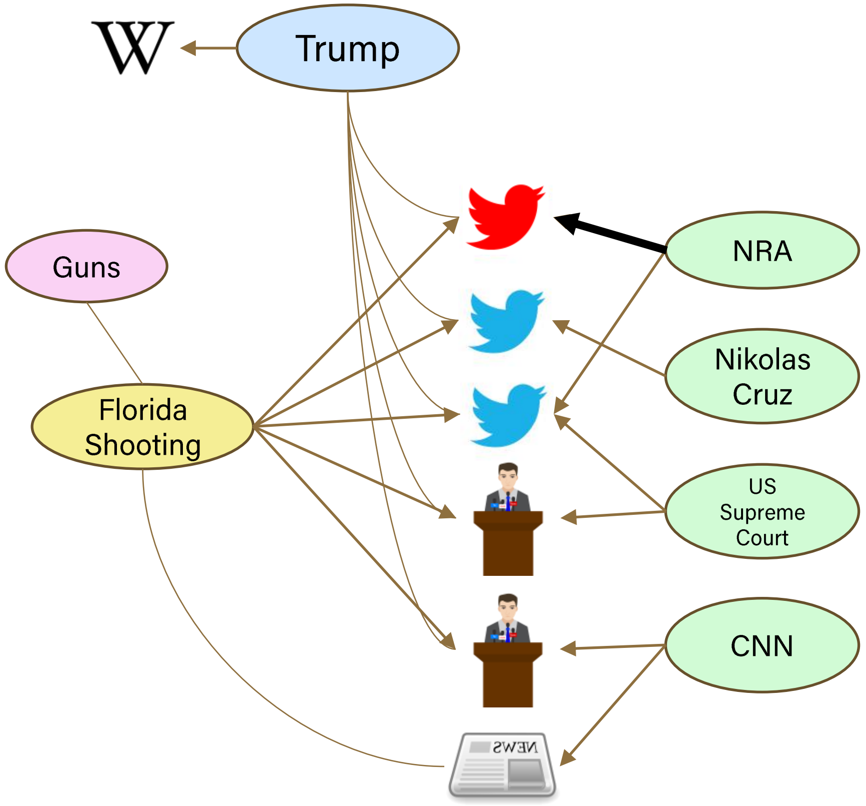


REFERENCED ENTITY PREDICTION

- Intuition behind this learning task is to enable our model to learn the correlation between the language used by a given politician in the context of a specific referenced entity

- We remove the link to the most frequent referenced entity in the graph and then mask all occurrences of that entity in the text

- We ask the binary question: Is the masked entity in this document ***NRA***?



LEARNING TASKS RESULTS

Model	IS Acc	IS F1	OS Acc	OS F1
Authorship Prediction				
BERT Adap.	93.01	92.31	95.56	95.20
Comp. Reader	99.49	99.47	99.42	99.39
Reference Entity Prediction				
BERT Adap.	76.57	75.21	76.26	73.67
Comp. Reader	78.52	77.51	78.98	78.62

Table 2: Learning Tasks results of baseline vs compositional reader. Accuracy and F1 score on the test set are Reported. IS denotes In-Sample performance (test authors are included in training set). OS denotes Out-of-Sample performance (train and test authors are mutually exclusive).

OVERVIEW

- Data
- Compositional Reader Pipeline
- Baseline Model
- Learning Tasks
- **Evaluation Tasks**
- Quantitative Evaluation Results
- Qualitative Visualization
- Conclusion

EVALUATION STRATEGY

- The aim of our strategy is to evaluate how well our representations reflect the actual meaning of the text in its real-world context.
- We design multiple quantitative tasks and qualitative visualizations to verify our hypothesis:
  1) Political Grade Data Alignment and Prediction
  2) Roll-Call Vote Prediction
  3) PCA visualizations of individual stances and group stances
  4) Opinion Descriptor Generation
  5) Disambiguation of vague, opinionated text

POLITICAL GRADES DATA

- Several organizations such as National Rifles Associations (NRA), League of Conservation Voters (LCV), Planned Parenthood etc., release issue-specific ratings of US politicians based on their voting patterns on the issue the organization cares about.
- Grades are indicative of the politicians' overall activity on the issue and how well it aligns with the grading organization's stance
- We design two evaluation tasks to test whether our representations are effectively able to:
  1) Align with the grades in a zero-shot setting (*Grade Paraphrase Task*)
  2) Predict the grades given politicians' public discourse (*Grade Prediction Task*)

GRADE PARAPHRASE TASK

- In this task, we evaluate the zero-shot language equivalence capabilities of our representations

- Grades are divided into two classes: grades from A+ to B- are in positive class and grades from C+ to F are clustered into negative. We formulate representative sentences for them. For example:
  1) POSITIVE: I strongly support the NRA
  2) NEGATIVE: I vehemently oppose the NRA

- We find the nearest class to the of each politician's issue representation based on their cosine similarity

GRADE PREDICTION TASK

- Given a politician's public discourse related to a specific issue, the aim of the task is to predict their organizational grade.

- For NRA data, it is designed as 5-class classification task for grades: A, B, C, D & F. We perform 10-fold cross-validation on the grade data.

- We compare the performance of the compositional reader representations to three strong baselines: pooled-BERT embeddings, BERT Adaptation baseline and Encoder embeddings.

- We design a similar evaluation task for LCV grades as well.

OVERVIEW

- Data
- Compositional Reader Pipeline
- Baseline Model
- Learning Tasks
- Evaluation Tasks

- **Quantitative Evaluation Results**

- Qualitative Visualization
- Conclusion

GRADE DATA RESULTS

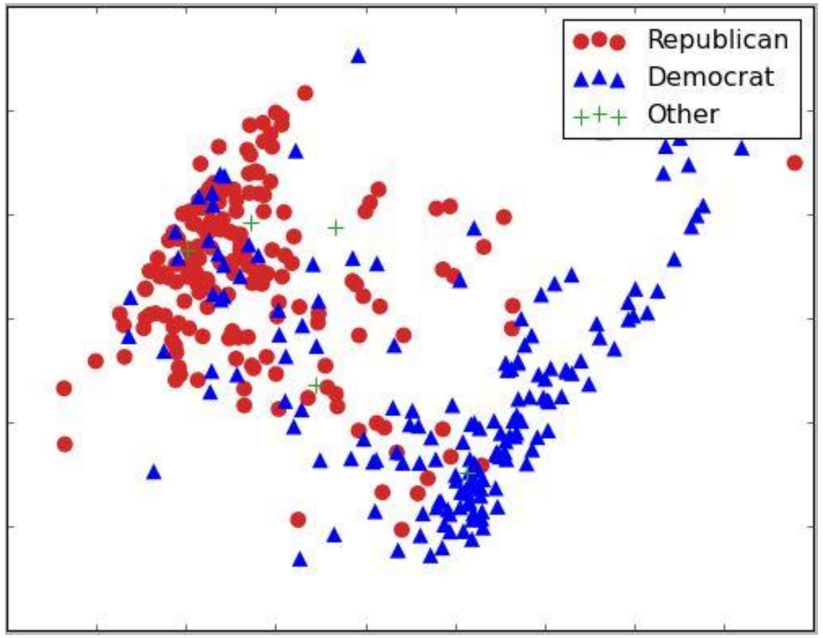
Model	PP (All)	PP (A/F)	NRA Val	NRA Test	LCV Val	LCV Test
BERT	41.55%	38.52%	55.93 ± 0.72	54.83 ± 1.79	54.28 ± 0.31	52.63 ± 1.21
BERT Adap.	37.54%	42.62%	71.23 ± 3.93	69.95 ± 3.33	60.58 ± 1.56	59.09 ± 1.77
Encoder	56.16%	48.36%	83.95 ± 1.24	81.34 ± 0.86	65.10 ± 0.46	63.42 ± 0.35
Comp. Reader	63.32%	63.93%	84.19 ± 0.98	81.62 ± 1.23	65.55 ± 1.33	62.24 ± 0.56

Table 3: Results of *'Grade Paraphrase'* and *'Grade Prediction'* tasks. Accuracy is reported. NRA and LCV denote respective Grade Prediction tasks. Mean ± Std. Dev for 5 random seeds for Grade Prediction showing statistical significance.

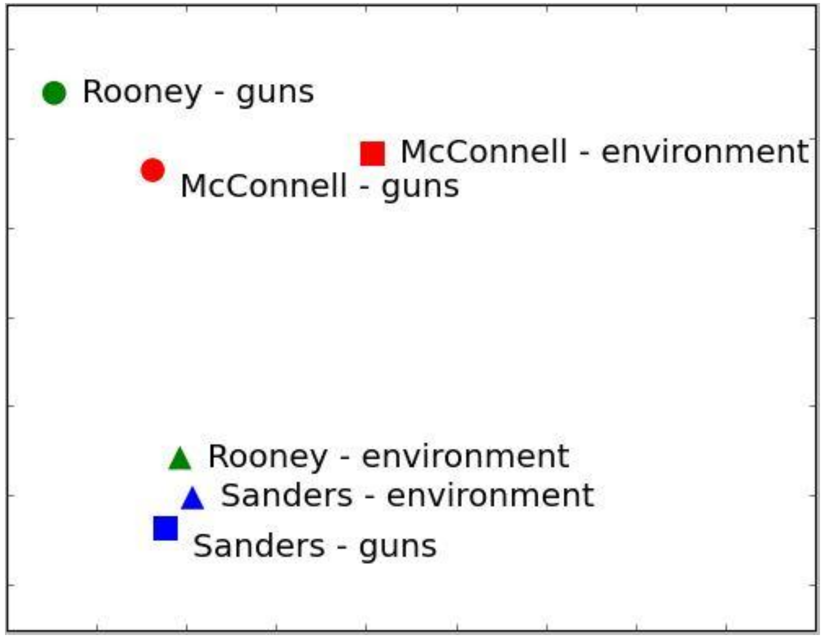
OVERVIEW

- Data
- Compositional Reader Pipeline
- Baseline Model
- Learning Tasks
- Evaluation Tasks
- Quantitative Evaluation Results
- **Qualitative Visualization**
- Conclusion

QUALITATIVE VISUALIZATIONS



(a) Issue '*Guns*'



(b) Individual Stances

PCA Visualizations of Politician Embeddings

OPINION DESCRIPTOR GENERATION

- In this evaluation, we identify the nearest adjectives in the embedding space to the summary embedding of a politician's stance on an issue

Issue	Opinion Descriptors	Issue	Opinion Descriptors
Mitch McConnell	Republican	Nancy Pelosi	Democrat
abortion	fundamental, hard, eligible, embryonic, unborn	*abortion*	future, recent, scientific, technological, low
environment	achievable, more, unobjectionable, favorable, federal	*environment*	forest, critical, endangered, large, clear
guns	substantive, meaningful, outdone, foreign, several	*guns*	constitutional, ironclad, deductible, unlawful, fair
immigration	federal, sanctuary, imminent, address, comprehensive	*immigration*	immigrant, skilled, modest, overall, enhanced
Donald Trump	Republican	Joe Biden	Democrat
guns	terrorist, public, ineffective, huge, inevitable, dangerous	*guns*	banning, prohibiting, ban, maintaining, sold
immigration	early, dumb, birthright, legal, difficult	*taxes*	progressive, economic, across-the-board, annual, top

Table 5: Opinion Descriptor Labels for Politicians. They show the most representative adjectives used by the politicians in context of each issue.

OVERVIEW

- Data
- Compositional Reader Pipeline
- Baseline Model
- Learning Tasks
- Evaluation Tasks
- Quantitative Evaluation Results
- Qualitative Visualization

- **Conclusion**

CONCLUSION

- We tackle the problem of *understanding politics*, i.e., creating unified representations of political figures capturing their views and legislative preferences, directly from raw political discourse data originating from multiple sources.

- We propose the Compositional Reader model that composes multiple documents in one shot to form a unified political entity representation, while capturing the real-world context needed for representing the interactions between these documents.

- This work is intended to inspire research in multiple directions:
  - Datasets that require an understanding of the real-world dynamics to be effectively solved
  - Towards more intricate computational models that cater for nuances of the challenge of understanding text in context
  - Towards design of meaningful learning tasks that help the model in capturing various aspects the text in context
  - Towards development of a thorough and exhaustive evaluation strategy for such models

ACKNOWLEDGEMENTS

We thank Shamik Roy, Nikhil Mehta and the anonymous reviewers for their insightful comments. This work was partially supported by an NSFCAREER award IIS-2048001.

THANK YOU!

Questions?