12^{th} International Conference on Computational Social Science IC^2S^2
July 28-31, 2026, Burlington, VT, United States

Learning Policy Processes from Social Media

Misha Melnyk[1], Mitchell Dolny[1], Joshua D. Elkind[1], A. Michael Tjhin[1], Saisha Chebium[1], Blake VanBerlo[1], Annelise Russell[2], Michelle M. Buehlmann[3], and Jesse Hoey[1]
[1] *Cheriton School of Computer Science, University of Waterloo*
[2] *Martin School of Public Policy & Administration, University of Kentucky*
[3] *Political Science & Public Administration, University of North Dakota*

Keywords: policy, transformer, social media, Twitter, US Senate

Extended Abstract

This study examines whether machine-learning models can reliably reproduce expert interpretations of political communication at scale. We study a large dataset of US senator postings on Twitter (1.68m tweets in total). Our objective is to develop an automated method to label senatorial posts as either in the problem or solution streams of Kingdon2003. Two academic policy experts (also co-authors) labeled a subset of 3967 tweets as either problem, solution, or other, achieving a high internal consistency. We split off a subset of 500 tweets into a test set, with the remaining 3467 used for training. During development, this training set was further split by 60/20/20 proportions for fitting, validation, and development test sets. We investigated supervised learning methods for building problem/solution classifiers directly on the training set, evaluating their performance in terms of F1 score on the validation set, allowing us to rapidly iterate through models and hyperparameters, achieving an average weighted F1 score of over 0.8 across the three categories using a BERT model.

Two authors, both experts in political science and the Kingdon model, independently labeled each tweet in the training set as "problem," "solution," or "other" and their ratings achieved an internal consistency of over 0.83 (Cohen's Kappa) and F1-macro 0.89. Tweets were first coded in a categorial fashion based upon whether the message includes any 1) problem-oriented or 2) solution-oriented language or 3) lacks either component. In training our models, we only use those tweets where the two labelers agreed on the rating unless otherwise noted.

Policy problem rhetoric captures how senators use Twitter to highlight issues, challenges, or shortcomings in governance. Messages were coded as policy problem mentions if they explicitly identified a social, economic, or political issue in need of attention or reform. These tweets range from broad problem statements (*"families are struggling to afford child care"*) to more specific critiques of existing policy (*"SNAP funding needs to be increased"*). Policy solution rhetoric includes messages that emphasize proposed actions, reforms, or achievements aimed at addressing specific policy issues. Tweets were coded as policy solution mentions when they highlighted concrete steps, such as introducing legislation, implementing programs, or celebrating successful policy outcomes. These messages often underscore a senator's role in advancing solutions or align their work with broader policy goals (*"I introduced a bill to lower prescription drug costs for seniors"*). In many cases, such tweets serve to demonstrate responsiveness and action. The non-problem/solution tweets — typical non-policy tweets altogether - are those messages that have no identifiable mentions of either a problem/solution. These messages are most commonly birthday messages, greetings, and mentions of constituent meetings, e.g. *"Allow me to welcome all of the new members who were just sworn in."*

We looked at three kinds of models: Logistic Regression [1], XGBoost trained decision trees [6], and BERT architecture neural networks [2]. Logistic Regression and XGBoost Trees

use just the year and author and serve as baselines, while BERT uses just the tweet text. These models either give a prediction directly or by outputting a confidence score. A confidence score assigns a value between 0 and 1 to each label, analogous to a probability. The sum of the confidence scores is 1. In the following we use BERT-base and BERT-large, see [5] for more details. Figure 1 is a schematic of the model showing the various layers.

In total, 20 trials are run of the main evaluation training. We also did a 7-fold cross validation, in which 140 runs are done of cross validation, 7 folds in each of 20 trials. The hyperparameters and configuration used is shown in Table 1. All other parameters were default, either defined by PyTorch or PyTorch Lightning.[1]

We also used a semi-supervised boosting method which trains a supervised model on the labeled data (as in the last section) and then uses it to assign labels with confidences to every unlabeled example [3]. Those labels with the highest confidence are then used to fine-tune the model. Finally, two LLM-based unsupervised (zero-shot using GPT-4.0 via the Martian API) methods were also attempted. Each of these has a mechanism for using data without labels. We found that modern "AI" models were capable of classifying text according to Kingdon's model without having the model explained. We investigated using an LLM directly to classify each tweet as problem/solution/political ("direct method"), and also using it to give confidences in each of the three categories, which are then optimized over using the VALIDATION set ("confidence method"). We evaluate two prompt variants: one requiring a class label with a brief explanation, and one requiring only the class label. We call these LLM based methods "unsupervised" except one must be aware that the LLM training data included (1) many documents and descriptions of the Kingdon model; and (2) the exact tweet being analyzed as well as any subsequent responses (very likely). Prompting details can be found in [5]. For the pseudo-label experiments, the "disagreed" tweets were also used.

As shown in Table 2, transformer-based models substantially outperform classical machine-learning baselines. Across evaluation settings, BERT models consistently achieve weighted F1 scores exceeding 0.80, substantially outperforming two classical baselines and being comparable to human rater agreement. Standard deviations across 20 random trials are less than 0.01 on accuracies and F1 scores. This suggests that the text does have a signal that differentiates it between these classes. Performance is comparable across annotation variants, with the BERT base (BERTweet) configuration achieving the strongest and most consistent overall performance. The pseudo-labeling strategy achieves the highest accuracy and macro F1 score, indicating stronger agreement with expert annotations when extending classification beyond the labeled dataset. In contrast, LLM-based approaches show substantially lower macro F1 scores. Recognizing that these LLMs are continuously improving, and that there may be other more effective prompting strategies, iterative pseudo-labeling using BERT-large (or small depending on resources) was the training strategy used for large-scale classification of the full dataset.

We next present descriptive statistics derived from the fully labeled dataset. These analyses are exploratory and intended to illustrate the kinds of substantive questions that the dataset enables rather than to test specific hypotheses. Figure 2 the number of tweets per month by party. We can see the steep rise starting between 2010-2012 when Twitter was growing in popularity. Figure 3 shows the distribution of labels per month for each party. One thing that stands out is the increase in democrat "problem" tweets in 2017, DJT first term, and the mirror image in republican tweets in 2021, Biden first term. Solution tweets also see a bit of a surge in 2017. Figure 4 show shows the distribution by gender and by race. As above, things change in 2017, with an increase in problem tweets starting in 2017, DJT first term. Further, non-whites increase in "other" 2012-2017 second Obama term. More results may be seen in [5] as well as

[1] Code and data is available at `cs.uwaterloo.ca/~jhoey/research/polytweets/`.

the full codebase and dataset to be made available.

In this paper, we investigated methods for labeling a large dataset of US senator social media posts (on Twitter). Based on a small set of labeled tweets, we were able to automatically label the remaining 1.68 million tweets with sufficient confidence. We showed some descriptive results of this automatic labeling, with some encouragingly clear transitions over time or "phase-changes" that seem correlated with US political elections.

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[3] S. Frisli. Semi-supervised self-training for covid-19 misinformation detection: analyzing twitter data and alternative news media on norwegian twitter. *Journal of Computational Social Science*, 8(39), 2025.

[4] John W. Kingdon. *Agendas, Alternatives, and Public Policies*. Addison-Wesley, 2003.

[5] Misha Melnyk, Mitchell Dolny, Joshua D. Elkind, A. Michael Tjhin, Saisha Chebium, Blake VanBerlo, Annelise Russell, Michelle M. Buehlmann, and Jesse Hoey. Learning of policy processes using congressional social media. *arXiv:2604.03247*, 2026.

[6] T. T. Chen and C C. Guestrin. Xgboost. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 785–794, 2016.

model_name	vinai/bertweet-base1U	The base model used
dropout_p	0.1	Dropout probability in training
trials	20	Number of different starting seeds
cross_val_folds	7	How many folds for k-fold cross validation
learning_rate	0.00003	Learning rate for optimizer
max_epochs	25	Maximum number of epochs to do before stopping
accumulate_grad_batches	1	How many batches to do before updating gradients
stopping_patience	3	Number of epochs before triggering early stopping
batch_size	64	Number of tweets in a mini-batch
global_seed	2025	Seed value for starting state

Table 1: . Parameters used while running the training trials.

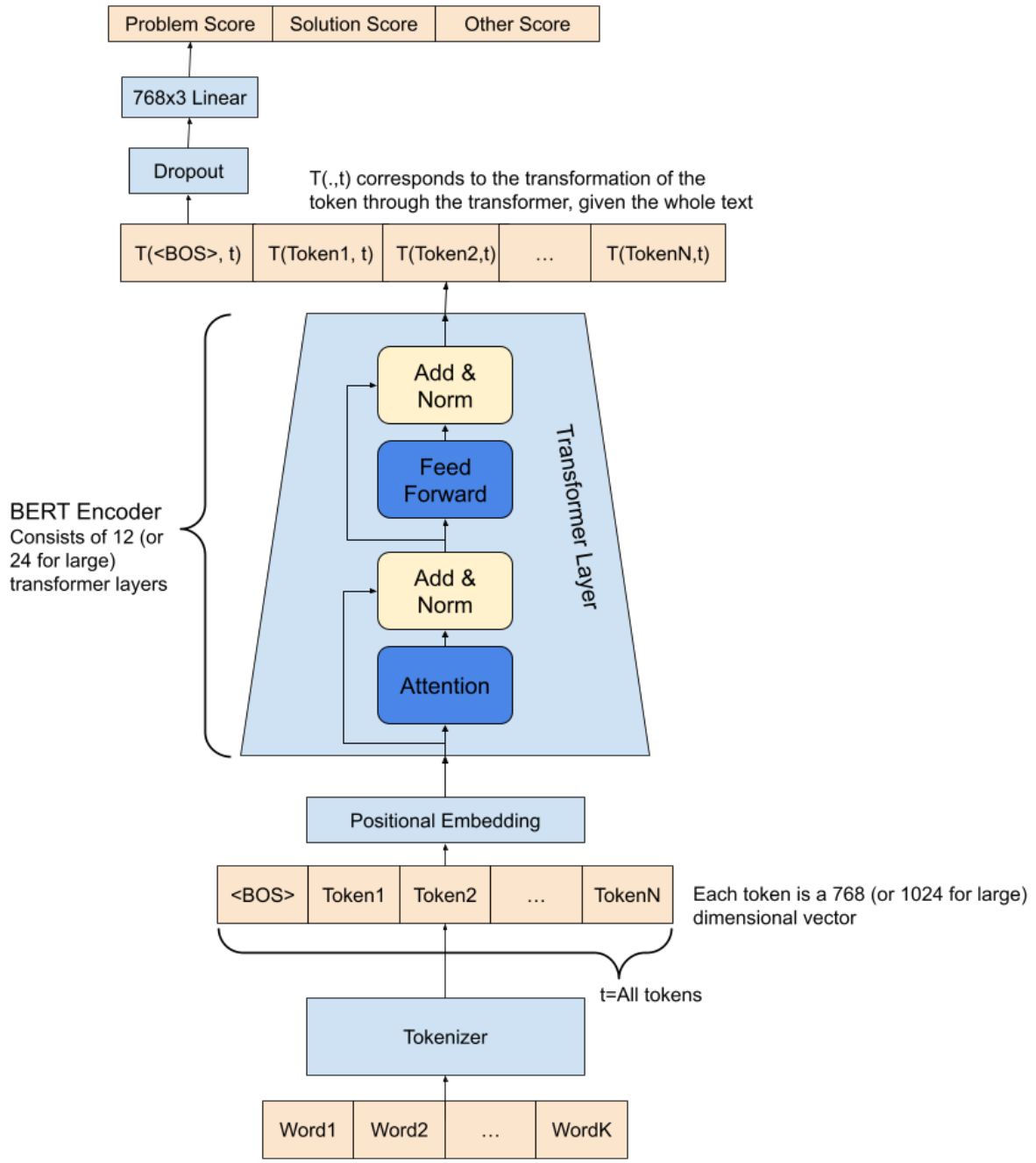

Figure 1: BERT Model Details showing flow of data from initial text to output scores. Information flows upwards in this picture, starting from a sequence of words (the Tweet) at the bottom. The Tokenizer separates out the raw text into tokens which are vectors in an embedding space. These vectors are augmented with positional information and fed into the Bidirectional Transformer (shown summarized as an attention layer and a feed forward layer in blue). The outputs of this model are then fed through a dropout and a linear layer to produce the final classification into problem/solution/other.

Classifier	Logistic Regression	XGBoost Trees	BERT		pseudo-label	LLM	
			base	large		direct	confidence
Avg F1	0.44	0.42	0.790	0.800	0.807	0.481	0.374
Avg Weighted F1	0.47	0.45	0.809	0.810	0.818	0.648	0.666
accuracy	-	-	0.793	-	0.816	0.658	0.609

Table 2: Performance comparison of classifier architectures on held-out evaluation data (macro F1 and weighted F1 scores). AR data used throughout.

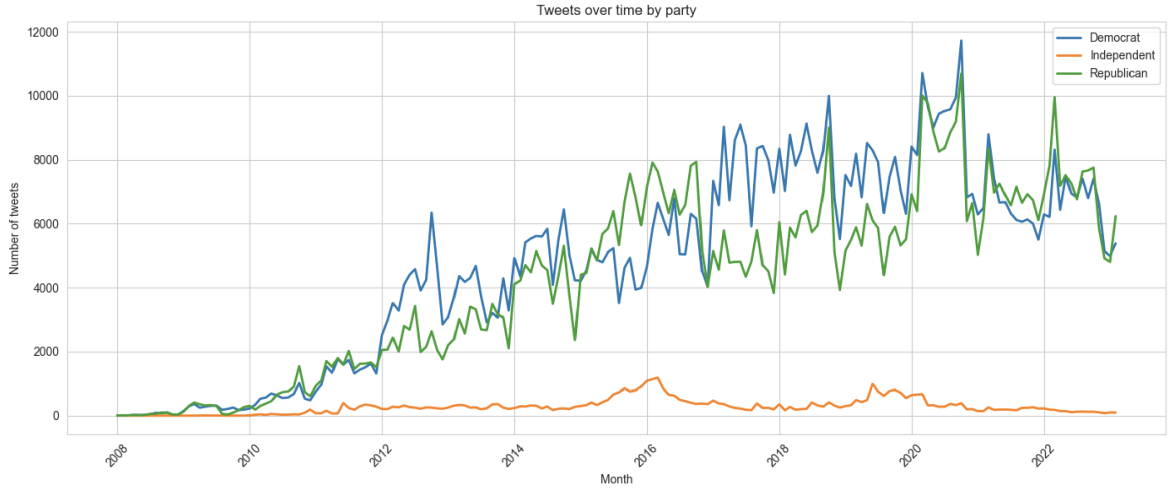

Figure 2: Tweets per month by party

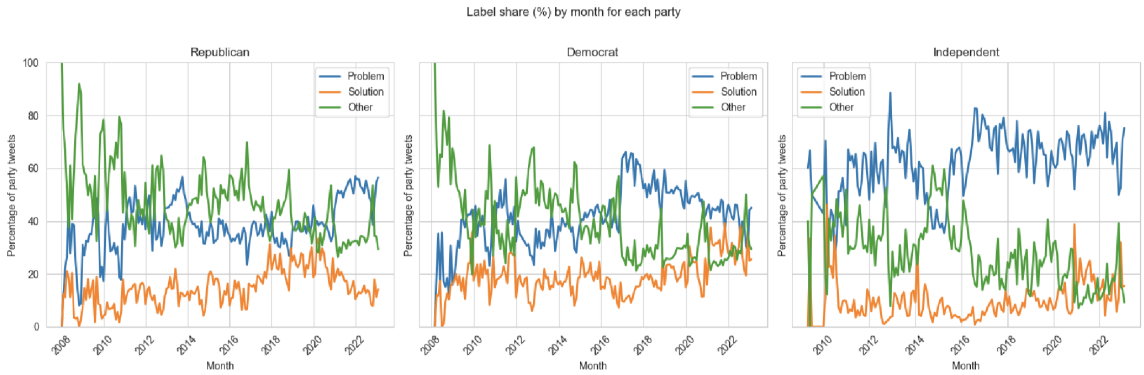

Figure 3: Distribution of labels per month for each party.

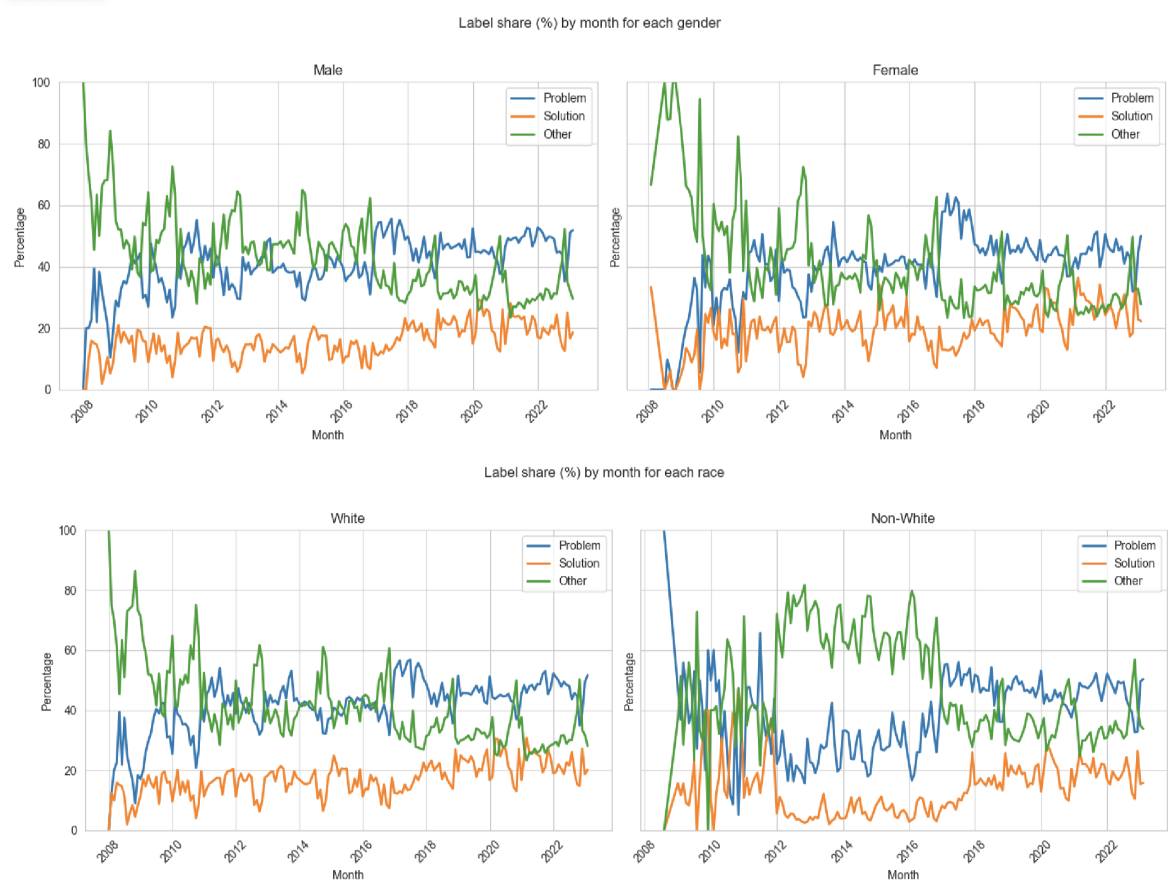

Figure 4: Distribution of labels per month by gender and race.