2017

Social Media in State Politics: Mining Policy Agendas Topics

Lei Qi
Iowa State University, leiqi@iastate.edu

Rihui Li
Iowa State University

Johnny S. Wong
Iowa State University, wong@iastate.edu

Wallapak Tavanapong
Iowa State University, tavanapo@iastate.edu

David A. M. Peterson
Iowa State University, daveamp@iastate.edu

Social Media in State Politics: Mining Policy Agendas Topics

Abstract
Twitter is a popular online microblogging service that has become widely used by politicians to communicate with their constituents.Gaining understanding of the influence of Twitter in state politics in the United States cannot be achieved without proper computational tools. We present the first attempt to automatically classify tweets of state legislatures (policy makers at the state level) into major policy agenda topics defined by Policy Agendas Project (PAP), which was initiated to group national policies.

Keywords
Policy agenda, Convolutional neural networks, Data augmentation

Disciplines
Computer and Systems Architecture | Data Storage Systems | Digital Communications and Networking | Models and Methods | Political Theory

Comments
This article is published as Qi, Lei, Rihui Li, Johnny Wong, Wallapak Tavanapong, and David AM Peterson. "Social Media in State Politics: Mining Policy Agendas Topics." (2017). doi: 10.1145/3110025.3110097. Posted with permission.

Social Media in State Politics: Mining Policy Agendas Topics

Lei Qi[1], Rihui Li[1], Johnny Wong[1], Wallapak Tavanapong[1], David A.M. Peterson[2]
[1]Department of Computer Science
[2]Department of Political Science
Iowa State University
Ames, IA 50011, USA
{leiqi, rihuili, wong, tavanapo, daveamp}@iastate.edu

Abstract— **Twitter is a popular online microblogging service that has become widely used by politicians to communicate with their constituents. Gaining understanding of the influence of Twitter in state politics in the United States cannot be achieved without proper computational tools. We present the first attempt to automatically classify tweets of state legislatures (policy makers at the state level) into major policy agenda topics defined by Policy Agendas Project (PAP), which was initiated to group national policies. We investigated the effectiveness of three popular machine learning algorithms, Support Vector Machine (SVM), Convolutional Neural Networks (CNN), and Long Short-Term Memory Network (LSTM). We proposed a new synthetic data augmentation method to further improve classification performance. Our experimental results show that CNN provides the best F1 score of 78.3%. The new data augmentation method improves the classification accuracy by about 2%. Our tool provides a good prediction of the top three popular PAP topics in each month, which is useful for tracking popular PAP topics over time and across states and for comparing with national policy agendas.**

Keywords: Policy agendas; Convolutional neural networks; Data augmentation

I. INTRODUCTION

Twitter has been widely used by politicians [1-2] to mine or guide the public opinion, learn users' political alignment by sentiment analysis of the users' tweets broadcast their political agendas. A policy agenda is a set of issues viewed as important by policy makers. Policy Agendas Project (PAP) [3] defines 20 major topics to trace changes in the national policy agendas and public policy outcomes. Congressional Bill Project [4] also categorizes congressional bills to PAP topics and subtopics manually. Automated classification of congressional bill titles into PAP topics has been investigated [5], but no similar techniques have been proposed for tweets.

Political science scholars are interested in a systematic approach to study the role of Twitter in state politics such as what topics state legislatures currently focus on, how policies are formed prior to bills (through Twitter?), how they are conveyed to the public via Twitter, how the conveyed agendas compared to the actual agendas of the state bills, what the top *k* policy agenda topics are in a given state during a given time period, how policy agendas diffuse among states, and how state agendas compared to the national agendas, just to name a few. Manual coding is time consuming and cannot keep up with tweets of members of the 50 state legislatures. Currently, we found close to 500 Twitter handles of these legislatures.

Automated classification of tweets into PAP topics is challenging because tweets are short and have informal content; some tweets are "ambiguous" to be assigned to one topic even for domain experts; it also exhibits class imbalance in which some PAP topics have many more tweets than the others.

Our contribution is the first investigation into this problem, the word-embedding data augmentation method, and the exploration of the effectiveness of the state-of-the-art machine learning algorithms on this classification problem. Our experiments reveal that the best classification algorithm offers about 78.3% F1 score [6]. The proposed data augmentation method improves the classification accuracy by about 2%. Our best classifier gives 91% prediction accuracy of the top three popular topics on tweets by Iowa and Nebraska legislatures during Jan. to Nov. of 2015.

II. RELATED WORK

Text classification has been widely studied in data mining, machine learning, databases, and information retrieval communities with applications in diverse domains. Prior to deep learning, SVM was shown to perform well for text classification with hand-crafted features [7]. CNN, adapted for text classification where the best features for classification are learned automatically during the training process, was shown to perform remarkably well [8].

The **role of social media in politics** has been studied. It was found that emotion dimensions typically reflect significant offline events and that average changes in Twitter mood levels were correlated with social, political, cultural, and economic events [9]. Sandberg et al. examined to what extent social media can possibly contribute in shaping the issue agenda regarding the political parties and identify what issues are salient in the online discussions during the European and Swedish 2014 elections [10].

Data augmentation has received more attention recently because supervised deep learning requires a large dataset to train a complex model with millions of parameters in general. Data augmentation methods can be divided into two categories depending on how the new samples are obtained. Real data augmentation draws real unlabeled samples not yet included the training dataset whereas synthetic data augmentation synthetically generates new training samples from the samples already in the training dataset. Synthetic data augmentation is typically done directly on images and

ASONAM '17, July 31-August 03, 2017, Sydney, Australia
 ACM ISBN 978-1-4503-4993-2/17/07..$15.00
http://dx.doi.org/10.1145/3110025.3110097

audio data (i.e., in data space). Unlike synthetic data augmentation for image and audio, a similar approach on text that affects the order of the words would change the semantic meaning of a sentence, which is not desirable. To the best of our knowledge, the only synthetic data augmentation method for text classification uses thesaurus [11].

III. METHODS

A. Hand-crafted Feature Extraction

For preprocessing, we investigate two scenarios—removal of stop words or not. We did not apply stemming since tweets are already short and different forms of a word may convey significantly different cues about the correct topics. We investigate both word-level and character-level n-grams; the character-level n-grams should better handle typographical errors commonly found in tweets better [12]. Finally, we extract Term Frequency-Inverse Document Frequency (TF-IDF) features on the n-grams [6].

B. Deep Learning Methods

We investigate two deep models: ***CNN*** and ***LSTM.*** We followed the CNN approach introduced in [8] as follows. Each tweet is first converted to a corresponding 2-dimensional matrix using Google word2vec [13]. Next, we apply convolutional operations on the matrix using hundreds of filters to get feature maps. Finally, we apply a max-pooling operation on the set of feature maps, which takes the maximum value as the feature value corresponding to each filter. Using different filters and window sizes, we obtain multiple features. These features are fed into a fully connected Softmax layer to obtain the final probability for each class; the label with the maximum probability is then chosen. As a sequence model, to discover long-term dependencies, we used the standard LSTM architecture [14].

C. Data Augmentation using Word Embedding

We explore a new data augmentation method using word embedding, which works as follows. First, the method randomly selects a word in a tweet. Let's call this word "anchor" word. It then makes k copies of the tweet by replacing the anchor word in each copy with each of the top k words that are closely similar to the anchor word in the given word embedding space. Unlike legislative bills, the language used in tweets even from state legislatures are not as formal as that used in the bills. Inevitably, some words such as hashtags, might not occur in the pre-trained word2vec model. For these words, we first apply word segmentation based on the intuition that hashtags are generally made up of meaningful words and the meaning of the hashtag is from these words. After segmentation, if the resulting word is still not in the pre-trained word2vec model, we keep the original word. Suppose all the words in the tweets have top k similar words, and the average length of a tweet is L; k^L new samples can be generated in theory.

We experiment with two sets of word embedding [13] [15] and investigate two methods to augment our original training dataset. One method randomly selects $p\%$ of the tweets that were generated using the aforementioned word embedding method and adds them to the original training set. In our experiment, we chose $p = 1$, (1% of the generated tweets) to minimize training time. Another method is aimed to compensate for the class imbalance problem. Let M be the maximum number of training samples of all the classes. We randomly add generated samples in each of the smaller classes to expand the class size to M.

IV. EXPERIMENTAL DESIGN AND RESULTS

A. Datasets

We collected tweets from a total of 472 Twitter accounts that include Senate and House official accounts for each state if exists and accounts of individual senators and house representatives for eleven states during the period of Jan. 1, 2009 to Nov. 29, 2015. We started labeling tweets of state legislatures from Iowa and Nebraska in this period; all the tweets in the dataset were labeled by two political science students according to PAP codebook under the guidance of the author who is a political science professor. Table I shows the number of tweets we used.

B. Experimental Design

We used scikit-learn [16] and the grid search method (with linear, poly, sigmoid, and RBF kernels) to find optimal parameters for the three investigated methods. We evaluated SVM performance using TF-IDF features computed from word and character level n-grams with and without removal of stop words as preprocessing. We used the CNN parameter values determined empiracally as follows: rectified linear units as the activation function, the drop out rate of 0.5, three window sizes (h) of 2, 3, 4 words with 150 feature maps each, and the mini-batch size of 128. We trained the CNN model using three variants (random, Static and Non-Static) [8] on the original and augmented datasets. For LSTM, we used the following parameters determined empirically: rectified linear unit as the activation function, the dropout rate of 0.2, the recurrent dropout rate of 0.2, and the mini-batch size of 32.

C. Experimental Result & Dicussion

C.1 Classification Effectiveness of CNN and LSTM

Figs. 1-2 show the CNN and LSTM performance on Iowa and Nebraska test datasets, respectively. CNN performs bettter than LSTM by about 7%. In addition, the non-static CNN scheme offers better performance than the static scheme because it continues updating the word embedding to fit this specific classification problem. CNN is able to mine more underlying patterns from short text data. As shown in Figs. 1-2, the augmented balance dataset using the proposed data augmentation method improves the CNN and LSTM preformance by around 2% because it not only generates more samples to compensate for a small training dataset, but also balances the dataset. Although LSTM was reported by some

TABLE I: Datasets of 21,336 tweets

States	Training Set	Test Set
Iowa	12461	1385
Nebraska	6741	749

researchers that it can can mining the long-term dependencies within sentence [14], it performs worse than CNN when dealing with short sentences in the experiments.

C.2 Classification Effectiveness of SVM

SVM accuracy, as shown in Fig. 3, is about 10% lower than that of the non-static CNN scheme. Fig. 3 shows that the SVM model trained on the balance augmented training dataset is most effective across all features and pre-processing techniques. The highest accuracy of 0.684 on the Iowa test dataset comes from using TF-IDF features of 6-ch grams (character n-grams) without removal of stop words (6-ch_N in Fig. 3) using the balance augmented training dataset. The suffix N in the technique names in Fig. 3 indicates no removal of stop words whereas the suffix Y indicates that stop word removal was used. About 1-2% improvement in accuracy is obtained when using the balance augmented training dataset compared to when using the original imbalance training dataset. This shows that the classification performance might not be improved if we only increase training samples but do not solve the class imbalance problem. Futhermore, the features from the character-level grams show better accuracy than those from the word-level grams in most cases when facing informal text because character-level grams can tolerate some errors to some extent [12]. In addition, we found little difference in classification accuracy between using Google word2vec [13] and tweet word2vec [15] embedding. Due to space limitation, Fig. 3 only shows the best result of the two pre-trained word embedding models.

C.3 Prediction of Top K Topics

To evaluate whether the CNN method with the current classification accuracy is of any practical value for political science research, we compare the prediction by the CNN method for the top k topics (k=3 in our experiments) with the ground truth using the Iowa and Nebraska test datasets. We first trained one CNN model for each state on the tweets posted by the legislatures of that state from Jan. 2009 to Dec. 2014 using the non-static scheme and our word-embedding data augmentation method to create a balance training dataset as described in Section III since they offered the best performance. Then, we used the trained CNN classifier to classify the tweets posted during Jan. 2015 to Nov. 2015. After obtaining the predicted topic labels of the tweets, we ranked the topics by the number of tweets predicted in each topic and obtained the top 3 topics excluding topic 0 because the tweets in this topic were not easily classified into any one PAP topic even by the domain experts.

We considered two scoring criteria, **exact match** and **rough match**, to compute the top-3 topic prediction accuracy. Using the exact match criterion, both the predicted topics and the predicted order must match exactly with the ground truth. The rough match criterion only considers whether the prediction gives the right topics, but does not need the order of the top 3 topics to be correct. For example, the ground truth top 3 topics for March 2015 are topic 5, topic 1 and topic 3 in this order. The predicted top 3 topics in this month are topic 1, topic 5 and topic 3 in this order. Using the exact match criterion, the matching score in this month is 0 because the order is wrong, but with the rough match criterion, the matching score is 1 because the classifier still gets the right three topics. The accuracy for each criterion is averaged over the matching scores for all the months under the consideration. The closer the number to 1 the better the accuracy.

Table II depicts the prediction accuracy of the top 3 topics for Iowa and Nebraska. The prediction accuracy for Iowa using the rough match criterion is good at 0.91. We correctly predicted the correct top 3 topics for 10 months out of 11 months. When considering the exact match criterion, the prediction accuracy is 0.82. This accuracy is still pretty good. When looking closely at the mis-predicted topics or mis-predicted order (marked in red in Table II), the number of tweets in the ground truth for these topics are very close, less than 20 tweets apart. In other words, the mis-predicted topics or mis-predicted order may be interchangeable. For Nebraska, the prediction accuracy using the rough match criterion is also 91%, but is only 75% using the exact match criterion. We also found the top 1 topic was always predicted correctly for all the months in the two states. Therefore, if the 2nd or 3rd topic is not as important or if we want to know the hottest topics discussed by legislatures in a specific state and time period, our current classifier is sufficiently accurate.

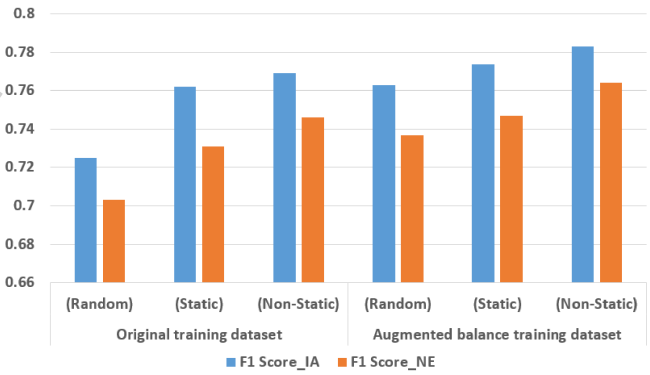

Fig. 1: F1 score of CNN models on the test dataset of Iowa & Nebraska

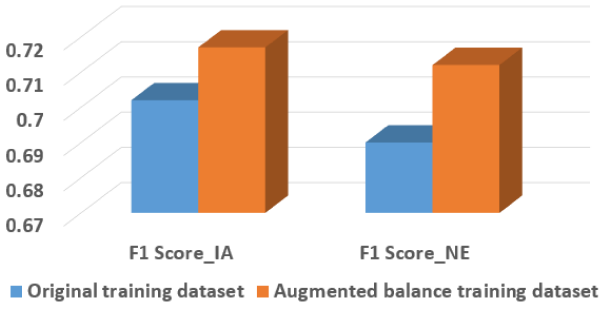

Fig. 2: F1 score of LSTM on the test dataset of Iowa & Nebraska

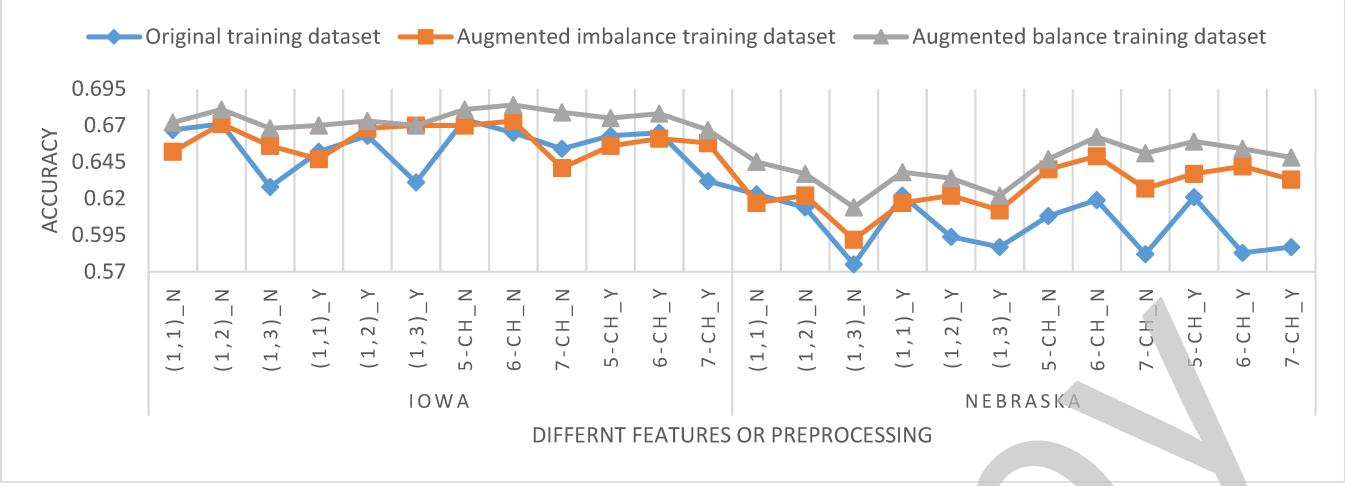

Fig. 3: Accuracy of SVM on Iowa and Nebraska test datasets

TABLE II: Accuracy of prediction of top 3 topics during Jan. 2015-Nov. 2015

State	Months	Jan.	Feb.	Mar.	Apr.	May	Jun.	Jul.	Aug.	Sept.	Oct.	Nov.	Accuracy
Iowa	Predicted topics	1,7,5	3,5,1	5,3,1	6,5,1	5,6,1	6,5,2	5,6,1	5,1,12	6,3,5	5,1,6	5,3,2	
	Ground truth topics	1,7,5	3,5,1	5,1,3	6,5,1	5,6,1	6,5,2	5,6,1	5,1,2	6,3,5	5,1,6	5,3,2	
	Rough match score	1	1	1	1	1	1	1	0	1	1	1	**0.91**
	Exact match score	1	1	0	1	1	1	1	0	1	1	1	**0.82**
Nebraska	Predicted topics	1,5,6	1,12,6	5,1,6	1,6,13	1,5,3	12,1,5	1,2,3	1,6,3	1,2,3	3,5,12	3,5,2	
	Ground truth topics	1,6,5	1,12,6	5,1,6	1,6,3	1,5,3	12,1,5	1,2,3	1,6,3	1,3,2	3,5,12	3,5,2	
	Rough match score	1	1	1	0	1	1	1	1	1	1	1	**0.91**
	Exact match score	0	1	1	0	1	1	1	1	0	1	1	**0.75**

V. CONCLUSION AND FUTURE WORK

We explore a new research problem of classification of a tweet into one of the pre-defined policy agenda topics used in several political science studies. We proposed a new word-embedding data augmentation method that generates synthetic data samples using pre-trained word embedding to address the class imbalance issue. We found that CNN provides significant improvement of about 10% over Support Vector Machine and about 7% over LSTM. The proposed data augmentation contributed to the improvement in classification accuracy. With the best CNN method in our experiments, we can estimate the top 3 hottest topics tweeted in a month in a given state quite reliably, which is potentially useful for testing different theories about the diffusion of policy agendas across states. We plan to investigate an active deep learning method to speed up the process of obtaining the ground truth by recommending the minimum number of tweets for the domain experts to label to achieve high prediction accuracy of the top-k topics.

REFERENCES

[1] Michael D. Conover, Bruno Goncalves, etal, Predicting the Political Alignment of Twitter Users, 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011, pp. 192-199.

[2] Jungherr, Andreas. "Twitter in politics: a comprehensive literature review." Browser Download This Paper (2014).

[3] Policy Agendas Project. Available: http://www.policyagendas.org

[4] E. S. A. a. J. Wilkerson. Congressional Bills Project. Available: http://congressionalbills.org/

[5] S. Purpura and D. Hillard, "Automated classification of congressional legislation," in Proceedings of Int'l Conf. on Digital government research, San Diego, California, USA, 2006, pp. 219-225.

[6] Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. Introduction to information retrieval. Vol. 1. No. 1. Cambridge: Cambridge university press, 2008.

[7] Thorsten Joachims, Text Categorization with Suport Vector Machines: Learning with Many Relevant Features, In Proc. of the European Conference on Machine Learning,1998, pp. 137-142.

[8] Y. Kim. Convolutional neural networks for sentence classification. In Proc. of EMNLP, 2014.

[9] Bollen, Johan, Huina Mao, and Alberto Pepe. "Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena." ICWSM 11 (2011): 450-453.

[10] Sandberg, et al.. "The social media election agenda: Issue salience on Twitter during the European and Swedish 2014 elections." Advances in Social Networks Analysis and Mining (ASONAM), IEEE, 2016.

[11] Xiang Zhang, Junbo Zhao, and Yann LeCun. "Character-level convolutional networks for text classification." In Advances in neural information processing systems, pp. 649-657. 2015.

[12] Qi, Lei, et al. "Automated Coding of Political Video Ads for Political Science Research." Multimedia (ISM), 2016 IEEE International Symposium on. IEEE, 2016.

[13] Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).

[14] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9, no. 8 (1997): 1735-1780.

[15] Godin, Fréderic, et al. "Multimedia lab@ acl w-nut ner shared task: named entity recognition for twitter microposts using distributed word representations." ACL-IJCNLP 2015 (2015): 146-153.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et al. "Scikit-learn: machine learning in python," Journal of Machine Learning Research, 12(Oct), pp. 2825−2830, 2011.