XXXXXX

Yicheng Pu
yp653@nyu.edu

Tianyu Wang
tw1682@nyu.edu

Abstract

1 Introduction

Twitter has been widely used as a source of public opinions. Thus with Twitter data, it is possible to gain some insights of the US Presidential election, whose results are strongly affected by public opinion. Another important factor of the election is geolocation, since the US presidential election is voted by electors from each state. Topic modeling has been a classical tool for analyzing opinions in text, but traditional LDA[1] doesn't include geographical factor. In this paper we chose the Author-Topic model(ATM)[7] in which each author is associated with a distribution of topics, and we replaced author with states. We would like to compare topic distributions over different states during 2016 election, and to test whether their variations reflect political leanings. As a comparison, we also used supervised FastText[4] model to predict twitter users' political leaning, and aggregate predictions for single users into predictions for states.

2 Related Work

The tremendous amount of active users in Twitter has contributed a lot of information. Many other works tried to infer the population attitudes with mainstream information instead of collecting public opinion query. In 2010, Brendan had explored the correlation between sentiment word frequencies and political opinion. [6] After that, Kazem enhanced Brendan study by introducing geographic tagged, and uncovered candidates popularities locally. Meanwhile, they had claimed the validation of predicting elections by using Twitter data.[3] Following works got better performance as the nature language processing techniques improve. [9][8]

3 Model

While most other works used supervised learning techniques, we chose the Author-topic model, a modification of Latent Dirichlet Allocation(LDA) model. We also applied FastText tothis problem as a comparison.

Traditional LDA model assumes that a document is composed of different topics, and each topic has a specific distribution of related words. It simulates the process of document generation by picking topics their prior distribution, then picking words according to the picked topic.

Here the Author-Topic Model(ATM) adds another latent variable a, which represents authors, into the original LDA model. And it adds another assumption that the authors of each document would also affect the topic assignment.

a_d: Authors for each document

x: Author of a given word

$\theta \sim \tilde{Dirichlet}(\alpha)$: Distribution over topics for each author

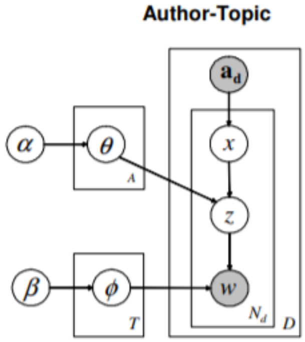


Figure 1: ATM

z: Topic assignment for each word

$\phi \sim Dirichlet(\beta)$: Word distribution for each topic w: Word

To explain the generative process more specifically, once a list of authors a_d are given, we uniformly select an author x for a_d. In our case, tweets only have a single user, so this step could be omitted. Then choose a topic z from x's specific topic distribution θ_a, which was generated from a Dirichlet prior. With z we could generte word w from z's specific word distribution ϕ_t, which was also generated from a Dirichlet prior.

After the generattive model is set up, we only need to infrence the optimal parameters out from their distributions. Here we used Gibbs Sampling[2] for inference as the original paper suggested [7].

4 Experiments

4.1 Data

We used the dataset provided by Harvard[5], which contains tweets collected during 3 presidential debates and election day. Since training ATM is highly time-consuming, and larger data size doesn't necessarily brings higher performance, we sampled 10% data from each dataset.

4.2 Data Cleaning

After data cleaning, we have 175,313 tweets left.

4.3 Author-Topic Model

4.3.1 Perplexity

We firstly used perplexity to measure the capacity of ATM model. Perplexity measures the difference between model's word distribution with real word distribution.

$$perplexity(w_d|a) = exp(-\frac{lnp(w_d|a_d)}{N_d})$$

We feeded the tweets and geographic labels into ATM, then tested the influence of number of topics for Perplexity. As shown in 2, the number of topics strongly affects Perplexity, increasing topics generally helps decreasing perplexity, but the curve converged around 300 topics.

4.3.2 methodology

After we get the 300 topics for each state, we firstly divide the topic distribution of each state by their sum to get the normalized author-topic distribution matrix AT with shape (51,300), then extracted two hand-crafted features:

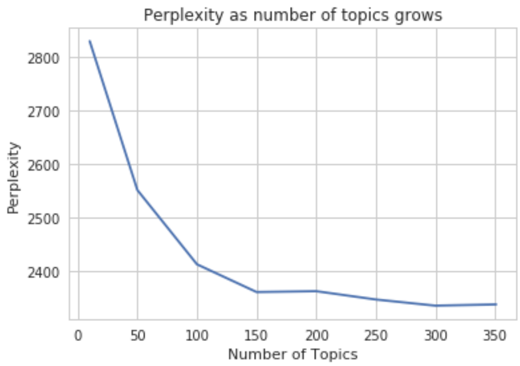


Figure 2: PP

1. red topics T_{red}: topics that have scores larger than mean+0.05*std in safe Republican states, but have scores lower than mean-0.05*std in safe Democrat states.
2. blue topics T_{blue}: topics that have scores larger than mean+0.05*std in safe Democrat states, but have scores lower than mean-0.05*std in safe Republican states.

Then we applied 12 different methods to predict swing states’ ranking, to illustrate them well, we have to make some notations first.

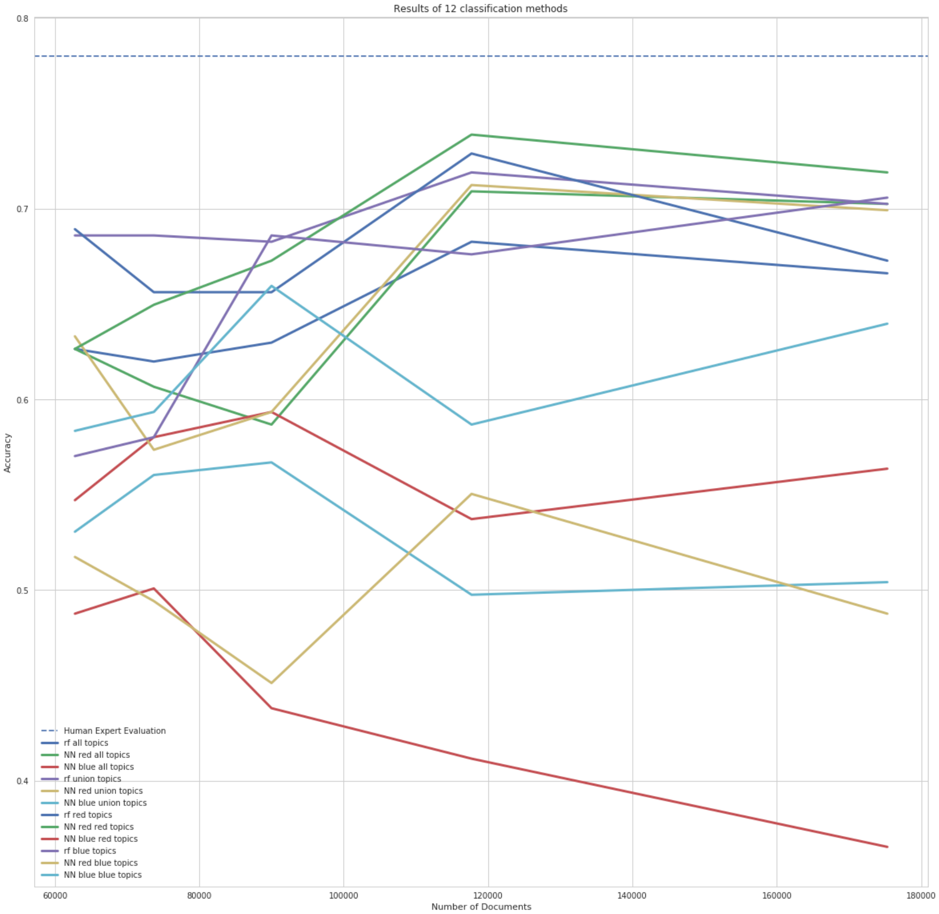
We denote Republican safe states’ author topic distribution by $AT_{rep,:}$, and Democrat safe states’ by $AT_{dem,:}$, swing states by $AT_{swing,:}$

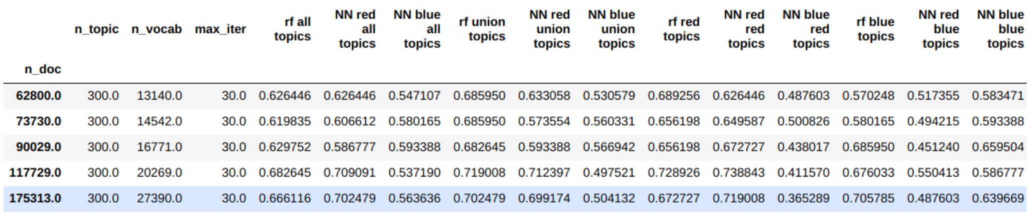
We also denote three main methodologies: f_{rf},f_{NN} and $f_{\sim NN}$:

1. $f_{rf}(X_{train}, X_{test})$ is a random forest classifier, which is trained on X_{train}, then predict X_{test} and rank the states in X_{test} by their predicted probability score.
2. $f_{NN}(target, X_{test})$ ranks states in X_{test} by their euclidean distance from the mean of target.
3. $f_{NN}(target, X_{test})$ ranks states in X_{test} by their negative euclidean distance from the mean of target.

Here are our 12 methods and their abbreviation:

1. ’rf all topics ’:$f_{rf}((AT_{rep,:}, AT_{dem,:}), AT_{swing,:})$
2. ’NN red all topics ’:$f_{NN}(AT_{rep,:}, AT_{swing,:})$
3. ’NN blue all topics ’: $f_{\sim NN}(AT_{dem,:}, AT_{swing,:})$
4. ’rf union topics ’: $f_{rf}((AT_{rep,T_{red} \cup T_{blue}}, AT_{dem,T_{red} \cup T_{blue}}), AT_{swing,T_{red} \cup T_{blue}})$
5. ’NN red union topics ’: $f_{NN}(AT_{rep,T_{red} \cup T_{blue}}, AT_{swing,T_{red} \cup T_{blue}})$
6. ’NN blue union topics ’: $f_{\sim NN}(AT_{dem,T_{red} \cup T_{blue}}, AT_{swing,T_{red} \cup T_{blue}})$
7. ’rf red topics ’: $f_{rf}((AT_{rep,T_{red}}, AT_{dem,T_{red}}), AT_{swing,T_{red}})$
8. ’NN red red topics ’: $f_{NN}(AT_{rep,T_{red}}, AT_{swing,T_{red}})$
9. ’NN blue red topics ’: $f_{\sim NN}(AT_{dem,T_{red}}, AT_{swing,T_{red}})$
10. ’rf blue topics ’: $f_{rf}((AT_{rep,T_{blue}}, AT_{dem,T_{blue}}), AT_{swing,T_{blue}})$
11. ’NN red blue topics ’: $f_{NN}(AT_{rep,T_{blue}}, AT_{swing,T_{blue}})$
12. ’NN blue blue topics ’: $f_{\sim NN}(AT_{dem,T_{blue}}, AT_{swing,T_{blue}})$





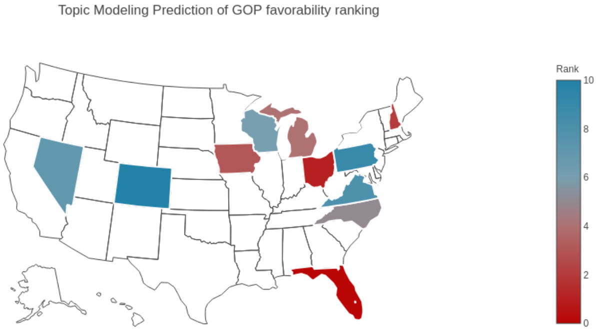
	n_topic	n_vocab	max_iter	rf all topics	NN red all topics	NN blue all topics	rf union topics	NN red union topics	NN blue union topics	rf red topics	NN red red topics	NN blue red topics	rf blue topics	NN red blue topics	NN blue blue topics
n_doc															
62800.0	300.0	13140.0	30.0	0.626446	0.626446	0.547107	0.685950	0.633058	0.530579	0.689256	0.626446	0.487603	0.570248	0.517355	0.583471
73730.0	300.0	14542.0	30.0	0.619835	0.606612	0.580165	0.685950	0.573554	0.560331	0.656198	0.649587	0.500826	0.580165	0.494215	0.593388
90029.0	300.0	16771.0	30.0	0.629752	0.586777	0.593388	0.682645	0.593388	0.566942	0.656198	0.672727	0.438017	0.685950	0.451240	0.659504
117729.0	300.0	20269.0	30.0	0.682645	0.709091	0.537190	0.719008	0.712397	0.497521	0.728926	0.738843	0.411570	0.676033	0.550413	0.586777
175313.0	300.0	27390.0	30.0	0.666116	0.702479	0.563636	0.702479	0.699174	0.504132	0.672727	0.719008	0.365289	0.705785	0.487603	0.639669

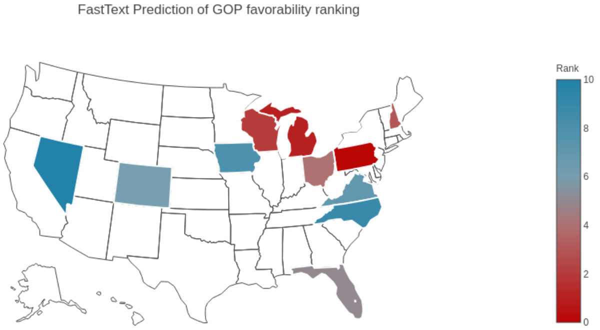
4.4 Fasttext

N-gram	Epochs	Training loss	Ranking ACC
1	20	0.18	0.57
1	25	0.17	0.59
1	30	0.11	0.54
2	20	0.16	0.59
2	25	0.09	0.59
2	30	0.11	0.57
3	20	0.14	0.59
3	25	0.10	0.57
3	30	0.1	0.60

Table 1: Table of FastText results

RED topics:





topic 290

'todddracula', 'dog', 'dtmag', 'vodka', 'likely', 'stare', 'celebrating', 'jbzfnzsc', 'correctly', 'casting', 'nba', 'pulled', 'protesters', 'pet', 'bitches', 'entry', 'idk', 'iwcdfytqf', 'podernfamily', 'overweight', 'delusional', 'music', 'vsfaoiioei', 'browns', 'gate', 'crap', 'reduced', 'wealthy', 'controlled', 'saudi'

topic 211

'rapes', 'phillyd', 'mike$_{p}ence','african','juanita','trash','pizza','cbsnews','crush','updates','knew','double','senat$

topic 176

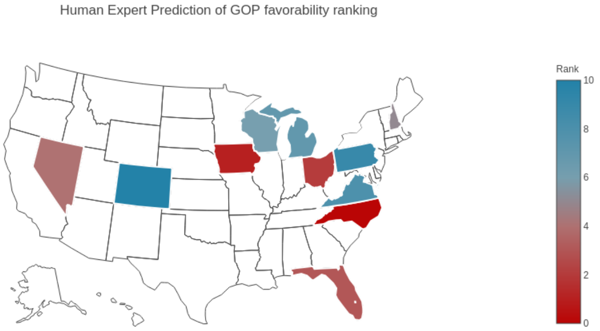
'murder', 'jill', 'spend', 'percent', 'skip', 'gregorybrothers', 'skin', 'covered', 'elementary', 'league', 'depresseddarth', 'killing', 'carter', 'ignorant', 'passing', 'forecast', 'increasing', 'stone', 'impression', 'green', 'markets', 'grew', 'sanity', 'annoying', 'factor', 'premiums', 'mood', 'yuge', 'refusing', 'stevenwhirsch'

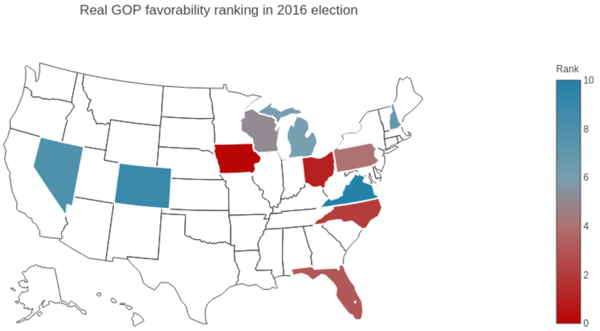
topic 33 'endorses', 'celebrity', 'arena', 'globeopinion', 'type', 'screen', 'catch', 'leaving', 'including', 'glass', 'greatest', 'muaspoeob', 'hair', 'citizens', 'term', 'vpdebate', 'worked', 'fool', 'analysis', 'english', 'scrum', 'included', 'adult', 'star', 'nafta', 'interested', 'badhombres', 'aka', 'mistake', 'ariannahuff'

topic 142

'worried', 'fix', 'machine', 'ones', 'text', 'trashvis', 'total', 'kzftvi', 'update', 'months', 'dems', 'prisonplanet', 'ugly', 'debaten', 'bloomberg', 'ran', 'should', 'helps', 'askaboutabortion', 'rqtvoii', 'greatest', 'agitators', 'loan', 'assholes', 'cards', 'pers', 'near', 'jerusalem', 'opinions', 'twice'

topic 112





’measured’, ’explain’, ’saudi’, ’yemen’, ’row’, ’celebrating’, ’mittromney’, ’drunks’, ’fawfulfan’, ’helpful’, ’nrulycw’, ’opposed’, ’noted’, ’philosoraptor’, ’busters’, ’kailijoy’, ’huma’, ’losses’, ’bought’, ’hungry’, ’bottles’, ’lesson’, ’metro’, ’investigation’, ’angela$_{r}ye',' controlled',' whining',' pussies',' employers',' roll'$

5 Conclusion

In conclusion, we found that Author-Topic model worked pretty well on this task, with stable performance approaching human expert’s prediction. And since we are lack of high-quality labeled data, unsupervised ATM outperformed supervised FastText, which indicates it could be applied in simiar scenearios in the future.

6 Future Work

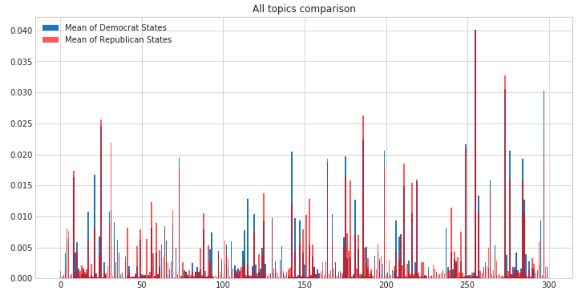
References

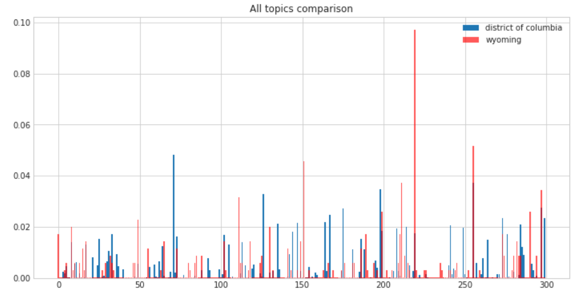
[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[2] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[3] K. Jahanbakhsh and Y. Moon. The predictive power of social media: on the predictability of us presidential elections using twitter. *arXiv preprint arXiv:1407.0622*, 2014.

[4] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.





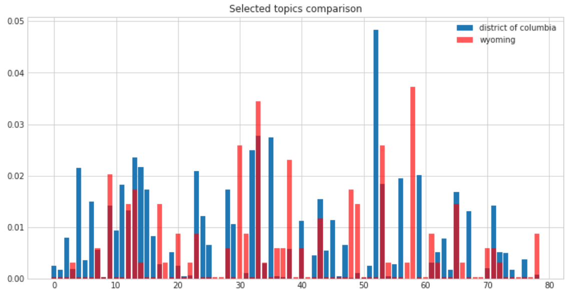
[5] J. Littman, L. Wrubel, and D. Kerchner. 2016 united states presidential election tweet ids, 2016.

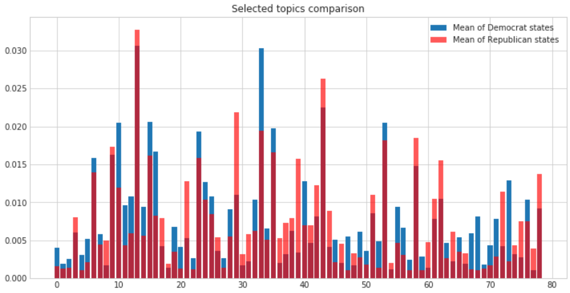
[6] B. O'Connor, R. Balasubramanyan, B. R. Routledge, N. A. Smith, et al. From tweets to polls: Linking text sentiment to public opinion time series. *Icwsm*, 11(122-129):1–2, 2010.

[7] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494. AUAI Press, 2004.

[8] E. Tunggawan and Y. E. Soelistio. And the winner is…: Bayesian twitter-based prediction on 2016 us presidential election. In *Computer, Control, Informatics and its Applications (IC3INA), 2016 International Conference on*, pages 33–37. IEEE, 2016.

[9] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan. A system for real-time twitter sentiment analysis of 2012 us presidential election cycle. In *Proceedings of the ACL 2012 System Demonstrations*, pages 115–120. Association for Computational Linguistics, 2012.


Selected topics comparison
🟦 district of columbia
🟥 wyoming
0.05
0.04
0.03
0.02
0.01
0.00
0
10
20
30
40
50
60
70
80


Selected topics comparison
🟦 Mean of Democrat states
🟥 Mean of Republican states
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
10
20
30
40
50
60
70
80