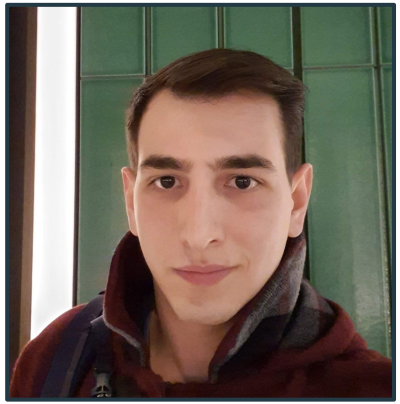
Improved Topic modeling in Twitter through Community Pooling

Federico Albanese and Esteban Feuerstein

(falbanese@dc.uba.ar)
@f__albanese
fedealbanese

SPIRE 2021



Federico Albanese

Esteban Feuerstein

Instituto de Ciencias de la Computación (ICC), UBA-CONICET
Instituto de Cálculo (IC), UBA-CONICET

Introduction

- **Characterizing the content of the messages** becomes vital for tasks like fake news detection, public opinion monitoring or personalized message recommendation.
- **Twitter posts are short and often less coherent** than other text documents, which makes it challenging to apply text mining algorithms efficiently.
- Tweet-pooling (**aggregating tweets into longer documents**) improves topic decomposition, but the performance varies depending on the pooling method.

Latent Dirichlet Allocation (LDA)

Steps to “generate” a document:

1) To generate a document D: asume a distribution of words for each topic.
   For example 50% a word w about football and 50% about politics.

2) Select a number N (the number of words in document D.
   For example 5.

3) Select N words for each document based on that probability.
   Example:
   D = ball, match, elections, candidate, gol.

(LDA does not care about the order of words)

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993-1022.

Latent Dirichlet Allocation (LDA)

Steps to find the topics:

1) Define the number of topics K.
2) Randomly assign each word w to one of the k topics
3) Iterate for each document d, for each word w in d:
   reassign the probability of word w to belong to topic t based on:

   $$p = p(\text{topic } t \mid \text{document } d) * p(\text{word } w \mid \text{topic } t)$$

4) Repeat step 3.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993-1022.

Latent Dirichlet Allocation (LDA)

Hipótesis:

1) The number of words in a document N ~ Poisson(**ξ**)
2) Proportion of each topic in a document **ϴ** ~ Dirichlet(α)
3) For each word w, it belongs to a topic t ~ Multinomial(**ϴ**)

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993-1022.

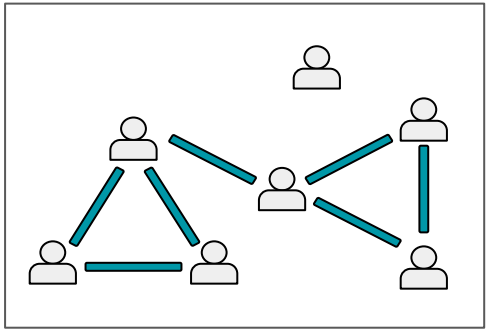
Latent Dirichlet Allocation (LDA)

Previous pooling models:

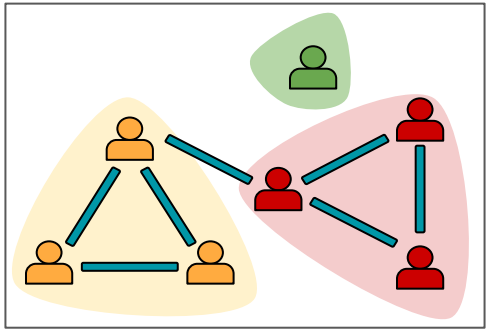
- Unpooled
- Author-Pooling
- Hashtag pooling
- Conversation pooling
- Network-based pooling

Main idea: Community pooling

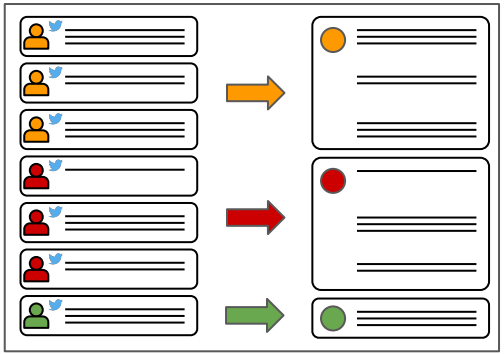
1) Build the retweet network.



2) Detect the communities using the Louvain method.



3) Aggregate the tweets based on the communities.



Our method makes the number of words in a document bigger and the number of documents smaller, and, therefore, a **more dense word co-occurrence matrix**, which has been shown to be beneficial to Latent Dirichlet Allocation (LDA).

Experiments

Datasets:

We constructed two datasets using the Twitter Streaming API:

- Generic Dataset: 115,359 tweets from December 15th to December 16th,2020.
    - music (36.78%),
    - family (23.94%),
    - health (17.21%),
    - business (14.90%),
    - movies (4.70%),
    - sports (2.44%).
- Event Dataset: 328,452 tweets from January 20th, 2021 (President Biden inauguration day).
    - Biden (69.45%),
    - joebiden (21.75%),
    - kamala harris (4.74%),
    - inauguration2021 (4.04%).

Experiments

Evaluation:

Experiments

Evaluation:

- Purity

$$\text{Purity}(T, Q) = \frac{1}{|T|} \sum_{i \in \{1 \ldots |T|\}} max_{j \in \{1 \ldots |Q|\}} |T_i \cap Q_j|$$

Experiments

Evaluation:

- Purity
- Normalized Mutual Information (NMI)

$$\mathrm{NMI}(T, Q) = \frac{2I(T,Q)}{H(T)+H(Q)}$$

Experiments

Evaluation:

- Purity
- Normalized Mutual Information (NMI)
- Supervised machine learning classifying task

  Split the dataset in train and test. Train a Naive bayes classifier on the first one and evaluate it on the second one, using the query of the tweet as a label.

Experiments

Evaluation:

- Purity
- Normalized Mutual Information (NMI)
- Supervised machine learning classifying task
- Document retrieval task

  For each tweet in a test set, retrieved the top 10 most similar tweets (using cosine similarity between the embeddings). Then measuring F1 to see if the labels match.

Experiments

Evaluation:

- Purity
- Normalized Mutual Information (NMI)
- Supervised machine learning classifying task
- Document retrieval task
- Running time

Experiments

Scheme	# of docs		Max # of words/doc		Mean # of words/doc	
	generic	event	generic	event	generic	event
Unpooled	115,359	328,452	783	1,023	137	128
Author	36,526	87,883	36,029	11,240	369	273
Hashtag	34,624	59,388	820,689	3,736,132	8295	173952
Conversation	35,484	67,276	12,480	41,024	141	130
Network-based	36,882	88,314	59,195	90,391	385	277
Community	24,657	31,303	2,077,085	5,284,617	874	1379

Results

Scheme	Purity		NMI		Classification		Retrieval		Running time	
	generic	event	generic	event	generic	event	generic	event	generic	event
Unpooled	0.664	0.733	0.436	0.110	0.814	0.843	0.837	0.893	137	388
Author	0.696	0.736	0.374	0.149	0.798	0.859	0.839	0.900	429	926
Hashtag	0.724	0.719	0.383	0.066	0.779	0.762	0.839	0.869	1,737	17,758
Conversation	0.658	0.733	0.436	0.110	0.814	0.843	0.835	0.908	738	1,569
Network-based	0.695	0.736	0.372	0.149	0.798	0.859	0.840	**0.910**	1131	2,841
Community	**0.780**	**0.779**	**0.439**	**0.310**	**0.827**	**0.889**	**0.843**	0.868	141	340

Our Proposed scheme, Community pooling, **outperformed all other schemes in most tasks and datasets.**

Conclusions

- The results on two heterogeneous datasets indicate that the novel Community based pooling outperforms all other pooling strategies in all tasks and metrics, with the only exception of the retrieval task on the event dataset.

- The running time analysis shows that Community pooling has a significant improvement in time performance in comparison with previous pooling methods, due to its capacity of reducing the total number of documents.

Community pooling

If you want to:

- know the topics of discussion in a twitter dataset,
- personalize message recommendations,
- group tweets based on their content,
- generate text embeddings based on the topics for a ML task

You could use this method. Code and full article:

https://github.com/fedealbanese/Community_pooling

Hope you liked our work. Feel free to talk to me if you want to 😃(falbanese@dc.uba.ar).

References

[1] Blondel, V. D., Guillaume, J.L., Lambiotte, R., and Lefebvre, E. Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment,2008(10): P10008, 2008

[2] Albanese, F., Lombardi, L., Feuerstein, E., and Balenzuela, P. Predicting shifting individuals using text mining and graph machine learning on twitter. arXiv preprint arXiv:2008.10749, 2020.

[3] Alvarez-Melis, D. and Saveski, M. Topic modeling in twitter: Aggregating tweets by conversations. In Proceedings of the International AAAI Conference on Web and Social Media, volume 10, 2016.

[4] Hong, L. and Davison, B. D. Empirical study of topic modeling in twitter. In Proceedings of the first workshop on social media analytics, pp. 80–88, 2010.

[5] Mehrotra, R., Sanner, S., Buntine, W., and Xie, L. Improving lda topic models for microblogs via tweet pooling and automatic labeling. InProceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, pp. 889–892, 2013.

[6] Ollagnier, A. and Williams, H. Network-based pooling for topic modeling on microblog content. In International Symposium on String Processing and Information Retrieval, pp. 80–87. Springer, 2019.