Network-Based Pooling for Topic Modeling on Microblog Content

Anaïs Ollagnier[1] [0000−0002−4349−5678] and Hywel Williams[1] [0000−0002−5927−3367]

Computer Science, University of Exeter, Exeter EX4 4QE, UK
{a.ollagnier,h.t.p.williams}@exeter.ac.uk

Abstract. Topic modeling with tweets is difficult due to the short and informal nature of the texts. Tweet-pooling (aggregation of tweets into longer documents prior to training) has been shown to improve model outputs, but performance varies depending on the pooling scheme and data set used. Here we investigate a new tweet-pooling method based on network structures associated with Twitter content. Using a standard formulation of the well-known Latent Dirichlet Allocation (LDA) topic model, we trained various models using different tweet-pooling schemes on three diverse Twitter datasets. Tweet-pooling schemes were created based on mention/reply relationships between tweets and Twitter users, with several (non-networked) established methods also tested as a comparison. Results show that pooling tweets using network information gives better topic coherence and clustering performance than other pooling schemes, on the majority of datasets tested. Our findings contribute to an improved methodology for topic modeling with Twitter content.

Keywords: Microblogs · LDA · Information Retrieval · Aggregation · User networks.

1 Introduction

Micro-blogging platforms such as Twitter have witnessed a rapid and impressive expansion, creating a popular new mode of public communication. Currently, Twitter has 6000 tweets written every second per day on average[1]. Twitter has become a significant source of information for a broad variety of applications, but the volume of data makes human analysis intractable. There is therefore considerable interest in adaptation of computational techniques for large-scale analyses, such as opinion mining, machine translation, and social information retrieval, among others. Application of topic modeling techniques to Twitter content is non-trivial due to the noisy and short texts associated with individual tweets. In the literature, topic models such as Latent Dirichlet Allocation (LDA) [1] or the Author Topic Model (ATM) [2] have proved their success in several applications (e.g. news articles, academic abstracts). However, results are more mixed when applied on short texts due to the data sparsity in each individual document.

[1] http://www.internetlivestats.com/twitter-statistics/ Date of access: 28th Jul 2019.

Several approaches have been proposed to design longer pseudo-documents by aggregating multiple short texts (tweets). Each document results from a pooling strategy applied in a pre-processing stage. In [3], an author-based tweet pooling scheme is used which builds documents by combining all tweets posted by the same author. A hashtag-based tweet pooling method is proposed by [4], which creates documents consisting of all tweets containing the same hashtag. The main goal behind these approaches is to improve topic model performance by training on the pooled documents, with efficacy measured against similar topic models trained on the unpooled tweets. Empirical studies with these approaches highlight inconsistencies in the homogeneity of generated topics. To overcome this problem, [5] propose a conversation-based pooling technique which aggregates tweets occurring in the same user-to-user conversation. This approach outperforms other pooling methods in terms of clustering quality and document retrieval. More recently, [6] propose to prune irrelevant tweets through a pooling strategy based on information retrieval (IR) in order to place related tweets in the same cluster. This method provides an interesting improvement in a variety of measures for topic coherence, in comparison to unmodified LDA baseline and a variety of other pooling schemes.

Several IR applications in context of microblogs use network representations [7] (e.g. document retrieval, document content). Here, we evaluate a novel network-based tweet pooling method that aggregates tweets based on user interactions around each item of content. Our intuition behind this method is to expose connections between users and their interest in a given topic; by pooling tweets based on relational information (user interactions) we hope to create an improved training corpus. To evaluate this method, we perform a comprehensive empirical comparison against four state-of-the-art pooling techniques chosen after a literature survey. Across three Twitter datasets, we evaluate the pooling techniques in terms of topic coherence and clustering quality. The experimental results show that the proposed technique yields superior performance for all metrics on the majority of datasets and takes considerably less time to train.

2 TWEET-POOLING METHODS

Tweet texts are qualitatively different to conventional texts, being typically short (≤ 280 characters[2]) with a messy structure including platform-specific objects (e.g. hashtags, shortened urls, user names, emoticons/emojis). In this context, tweet-pooling has been developed to better capture reliable document-level word co-occurrence patterns. Here, we evaluate four existing unsupervised tweet pooling schemes alongside our proposed network-based scheme:

Unpooled scheme: The default approach used as a baseline in which each tweet is considered as a single document.

Author pooling: Each tweet authored by a single user is aggregated as a single document, so the number of documents is the same as the number of unique users. This approach outperforms the unpooled scheme [9].

[2] In September 2017, Twitter expanded the original 140-character limit to 280 characters. See: `https://blog.twitter.com/official/en_us/topics/product/2017/tweetingmadeeasier.html`. Date of access: 11th Feb 2019.

Hashtag pooling: Tweets using similar hashtags are aggregated as a single document. The number of documents is equal to the number of unique hashtags, but a tweet can appear in several documents if it contains multiple hashtags. Tweets without hashtags are considered as individual documents. This method was shown [5] to outperform unpooled schemes. (Note that [4] showed improved performance by assigning hashtag labels to tweets without hashtags, but this technique adds computational cost and was not used here.)

Conversation pooling: Each document consists of all tweets in the corpus that belong to the conversation tree for a chosen seed tweet. The conversation tree includes tweets written in reply to an original tweet, as well as replies to those replies, and so on. Tweets without replies are considered as individual documents. In [5], conversation pooling outperforms alternative pooling schemes.

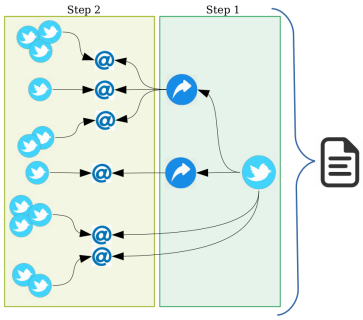

Fig. 1. Network-based tweet pooling. Each document is initialised with a seed tweet. In Step 1, the first layer of direct replies to the seed tweet are added. In Step 2, all tweets by users mentioned in the set of tweets resulting from Step 1 are also added.

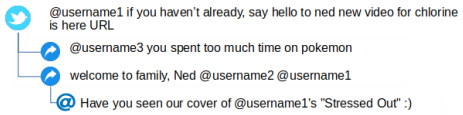

Fig. 2. Example content of a document created by network-based tweet pooling.

Network-based pooling: In this novel scheme, each document is aggregated from all tweets within the corpus that are associated with the seed tweet by a simple network structure (Figure 1 and Figure 2). In Step 1, tweets are aggregated that were written in reply to the seed tweet. In Step 2, we identify all mentioned users in the set of tweets from Step 1 (i.e. all users that are referenced in tweet text using the @ symbol). We then aggregate to the document all other tweets in the corpus that are authored by this user set.

This scheme differs from conversation pooling in two aspects. First, only direct replies are aggregated i.e. the first layer of replies from the conversation tree. Manual inspection of full tweet conversation trees showed that the conversation thread can shift in topic as the tree increases in depth. Use of the full tree can thereby capture topics which are not anymore related to those of the seed tweet. To identify reply tweets, we used the `in_reply_to_status_id` field returned by the Twitter API for each tweet. Second, exploiting tweets of all mentioned users allows the network-based pooling to access additional content from users interested in the topics of the original seed tweet. Leveraging this information, we construct a network based on both interactions and connections between users.

3 TWEET CORPUS BUILDING

Table 1. Distribution of latent categories in the datasets (labelled by search theme)

Dataset	No. of tweets	Category / % of Documents
Generic	658,492	Music/24.4 - Business/10.2 - Movie/18.5 - Health/14.7 - Family/7.4 - Sport/24.8
Specific	445,852	Arts&entertainment/9.7 - Business/12.4 - Law Enforcement&Armed Forces/6.2 - Science&technology/36.8 - Healthcare&medicine/25.5 - Service/9.4
Events	188,000	Natural disasters/37.1 - Transport/15.4 - Industrial/10.2 - Health/9.7 - Terrorism/27.6

To evaluate the portability of different pooling schemes we collected three tweet datasets with different levels of underlying thematic/topical heterogeneity. Data was collected using the public Twitter Search API[3] during 2018 and 2019. Each collection was created with a different list of API keywords and included tweets collected on different themes. For each chosen theme a list of terms was manually created. All tweets returned were collated in a single corpus, labelled by the theme. The three datasets collected were:

Generic. A wide range of themes. Tweets from 11 Dec'18 to 30 Jan'19 collected using keywords related to a range of themes ('music', 'business', 'movies', 'health', 'family', 'sports').

Event. Tweets from 23 Mar'18 to 22 Jan'19 associated with various events ('natural disasters', 'transport', 'industrial', 'health', 'terrorism'). Search terms were manually collated based on reading a sample of posts about disaster events.

Specific. Tweets from 21 Feb'18 to 11 Feb'19 associated with job adverts for different industries ('arts & entertainment', 'business', 'law enforcement & armed forces', 'science & technology', 'healthcare & medicine', 'service'). Search terms manually collated based on reading a sample of posts about job advertisements.

For each dataset, tweets retrieved by more than one query have been removed in order to preserve uniqueness of tweet labels. Table 1 illustrates the distribution of latent categories in each dataset. Each retrieved tweet was labeled according to a category corresponding to the query submitted. We leverage these labels to evaluate the topics produced by each model in term of clustering quality.

[3] `https://dev.twitter.com/rest/public/search`. Date of access: 19th Feb 2019.

4 EVALUATION METRICS

According to metrics used in previous studies [4,5,6], we evaluate models both in terms of clustering quality (purity and normalized mutual information (NMI)) and semantic topic coherence (pointwise mutual information (PMI)).

Formally, let T_i be the set of tweets assigned to topic i and let $T = \{T_1, \ldots, T_{|T|}\}$ be the set of topic clusters arising from a LDA model that produces $|T|$ topics. Then let L_j be the set of tweets with ground-truth topic j and let $L = \{L_1, \ldots, L_{|L|}\}$ be the set of of ground-truth topic labels with $|L|$ labels in total. Our clustering-based metrics are defined as follows:

Purity: Purity score is used to measure the fraction of tweets in each assigned LDA topic cluster with the true label for that cluster, where the 'true' label is defined as the most frequent ground-truth label found in that cluster. Formally:

$$Purity(T,L) = \frac{1}{|T|} \sum_{i \in (1,|T|)} \max_{j \in (1,|L|)} |T_i \cap L_j|$$

Higher purity scores indicate better reconstruction of the original 'true' topic assignments by the model.

Normalized Mutual Information (NMI): The NMI score estimates how much information is shared between assigned topics T and the ground-truth labeling L. NMI is defined as follows:

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

where respectively, $I(\cdot,\cdot)$ corresponds to mutual information and $H(\cdot)$ is entropy as defined in [8]. NMI is a number between 0 and 1. A score close to 1 means an exact matching of the clustering results.

Pointwise Mutual Information (PMI): The PMI score [10] evaluates the quality of inferred topics based on the top-10 words associated with each modeled topic. This measure is based on PMI which is computed as $PMI(u,v) = log(\frac{p(u,v)}{p(u)p(v)})$ where u and v are a given pair of words. The probability $p(x)$ is derived empirically as the frequency of word x in the whole tweet corpus, while probability $p(x,y)$ is the likelihood of observing both x and y in the same tweet. Coherence of a topic k is computed as the average score of PMI for all possible pairs of the ten highest probability words for topic k (i.e. $W_k = \{w_1, ..., w_{10}\}$). Formally:

$$PMI - Score(T_k) = \frac{1}{100} \sum_{i=1}^{10} \sum_{j=1}^{10} PMI(w_i, w_j)$$

where $w_i, w_j \in W_k$. Then coherence of a whole topic model is calculated as the average PMI-Score for all topics generated by the model.

5 Results

For each combination of the three datasets (Section 3) and five pooling schemes (Section 2), we calculated three evaluation metrics (purity scores, NMI scores and PMI scores; Section 4) by training LDA models with 10 topics.

Table 2 presents various statistics of the training sets obtained by applying the different pooling schemes. We filtered the datasets to keep only tweets written in English and those with more than three tokens. Tweets were converted to lowercase and all URLs, mentions (except with the network pooling scheme) and stop-words were removed. After the tokenization process, all tokens based only on non-alphanumeric characters (emoticons) and all short tokens (with < 3 characters) were also deleted. Test sets have been randomly extracted (30%) from each dataset preserving the same distribution of tweet categories. For each topic model we conduct five cross-validations.

Table 2. Corpus statistics.

Scheme	No. of documents			No. of tokens		
	general	specific	event	general	specific	event
Unpooled	658492	445852	188000	18991	14794	9454
Author Pooling	504253	340826	157377	18339	14091	9222
Conversation Pooling	649389	440682	185737	19301	15061	9668
Hashtag Pooling	585171	387522	174501	19868	15185	9348
Network Pooling	585171	402687	171266	19868	20065	13051

Table 3. Clustering metrics and coherence scores for different schemes and datasets.

Scheme	Purity			NMI			PMI Score		
	general	specific	event	general	specific	event	general	specific	event
Unpooled	0.396	0.316	0.220	0.176	0.108	0.058	−0.131	0.224	0.307
Author Pooling	0.377	0.399	0.326	**0.181**	0.176	0.124	0.892	−0.116	0.338
Conversation Pooling	0.341	0.359	0.310	0.136	0.141	0.110	−0.131	0.062	−0.131
Hashtag Pooling	0.337	0.250	0.245	0.145	0.045	0.071	0.293	0.347	**0.851**
Network Pooling	**0.418**	**0.503**	**0.362**	0.173	**0.228**	**0.155**	**0.912**	**0.582**	0.794

Table 3 summarises the average results obtained with each pooling scheme and dataset. According to the clustering evaluation metrics (purity and NMI), Network Pooling produced the best model performance on all datasets, with the exception of NMI scores on the General dataset, where it was narrowly outperformed by Unpooled and Author Pooling.

Results for other pooling schemes vary by metric and dataset. Author Pooling is the second-ranked scheme for most metrics/datasets, with Conversation Pooling also outperforming the Unpooled scheme in most cases. It is interesting to notify that Hashtag Pooling is mostly ineffective and gives performance worse than the baseline in most cases. This finding can perhaps be explained by the observation that hashtags are typically present in a minority of tweets (e.g. 19.6% of tweets have hashtags in the Specific dataset). Concerning the measure of the topic interpretability, coherence scores show

that the Network Pooling scheme gives better performance on all datasets, with the exception on the Event dataset, where it was narrowly outperformed by Hashtag Pooling.

6 Conclusion

Methods for aggregating tweets to form longer documents more amenable to topic modeling have been shown here and elsewhere to improve model performance. Here we have proposed a new network-based pooling scheme for topic modeling with Twitter data, that takes into account the network of users that engage with a particular tweet. Our approach improves topic extraction despite different levels of underlying thematic/topical heterogeneity of each dataset. While similar to conversation-based pooling in its use of reply tweets, the network approach includes otherwise un-linked content from users who authored replies. Experimental results showed that for the tests performed in this study, the network-based pooling scheme considerably outperformed other methods and was portable between datasets. Model outputs were improved on both clustering metrics (purity and NMI) and topic coherence (PMI).

Although the experiments presented have been conducted on the corpora collected on specific time intervals which reduces the shifting of conversation threads, especially when we collect documents authored by a cited user in response to the seed tweet. On a larger scale, topic shifting might be handled by adding conditions on document timestamps or topic correlation. In addition, the experimental findings suggest that network-based approaches might offer a useful technique for topic modeling with Twitter data, subject to further testing and validation with other datasets.

Acknowledgements This work was supported by the Institute of Coding which received funding from the Office for Students (OfS) in the United Kingdom.

References

1. Blei, D. M., Ng, A. Y., & Jordan, M. I.: Latent dirichlet allocation. In: Journal of Machine Learning Research, vol. 3, no Jan., pp. 993–1022. (2003).
2. Rosen-Zvi, M., Griffiths, T., Steyvers, M., & Smyth, P.: The author-topic model for authors and documents. In: Proceedings of the 20th conference on Uncertainty in artificial intelligence. AUAI Press. pp. 487–494. (2004).
3. Hong, L., & Davison, B. D.: Empirical study of topic modeling in twitter. In: Proceedings of the 1st workshop on Social Media Analytics. ACM. pp. 80–88. (2010).
4. Mehrotra, R., Sanner, S., Buntine, W., & Xie, L.: Improving lda topic models for microblogs via tweet pooling and automatic labeling. In: Proceedings of the 36th international ACM SIGIR conference on Research and Development in Information Retrieval. ACM. pp. 889–892. (2013).
5. Alvarez-Melis, D., & Saveski, M.: Topic modeling in twitter: Aggregating tweets by conversations. In: Proceedings of the 10th international AAAI conference on Web and Social Media. pp. 519–522. (2016).
6. Hajjem, M., & Latiri, C.: Combining IR and LDA topic modeling for filtering microblogs. In: Procedia Computer Science, vol. 112, pp. 761–770. (2017).

7. Ahmad, W., & Ali, R.: Information retrieval from social networks: A survey. In: Proceedings of the 3rd international conference on Recent Advances in Information Technology (RAIT). IEEE. pp. 631–635. (2016).
8. Manning, C., Raghavan, P., & Schtze, H.: Introduction to information retrieval. In: Natural Language Engineering, vol. 16, no 1, pp. 100–103. (2010).
9. Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E. P., Yan, H., & Li, X.: Comparing twitter and traditional media using topic models. In: European conference on Information Retrieval. Springer, pp. 338–349. (2011).
10. Lau, J. H., Newman, D., & Baldwin, T.: Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics. pp. 530–539. (2014).