Improved Topic Models For Social Media via Community Detection Using User Interaction And Content Similarity

Mehta Prateek, Varma Vasudeva
IIIT Hyderabad
Hyderabad,India
prateek.mehta@research.iiit.ac.in, vv@iiit.ac.in

***Abstract*—Topic models, such as Latent Dirichlet Allocation(LDA) have historically served as a successful tool for various data mining applications on conventional documents such as news articles or academic abstracts. However, standard use of topic models on social media posts pose several problems because social media posts are short, messy and generated non-uniformly by the users of the social media platforms. In this paper we propose a new approach of community based document pooling to train better topic models over social media posts and address these problems without modifying the basic machinery of LDA. We compare our approach to the popular user based pooling scheme and show significant improvement in the quality of topic models.**

I. Introduction

Social media posts have become a huge mine of data. They are one of the most extensively used medium of communication in modern times and hence prove to be a very crucial source of information for applications such as breaking news detection, trend analysis, sentiment analysis, recommendation systems, advertisements and others. A vast set of these applications require understanding of user interests. Salient patterns in content generated by the user serve as good indicators of user interest. One of the very important data mining tools which has been historically used to highlight topical patterns from standard text domains is probabilistic topic models. LDA(Latent Dirichlet Allocation) [1] is a generative, Bayesian, latent variable topic model. It is arguably the most common and simplest topic model. LDA has been very succesful in highlighting thematic topics in conventional text documents due to its ability to use word co-occurrence statistics [2]. Therefore, it has been successfully adapted to model diverse range of documents ranging from news articles, research abstracts, web-pages etc. However social media poses several challenges to standard application of LDA.

LDA is unsupervised in nature and only requires a training corpus. The simplest way to train LDA model for social media posts, as used for conventional documents, is to use all the posts generated on the social media platform as individual documents and use collection of posts as training corpus. However, content generated on social media, specially in short text environments like microblogging sites is characterized by a large number of short and noisy posts generated non-uniformly by a diverse set of users. This poses challenges in employing of LDA models to their full potential. Below are the two challenges that we target to resolve using our proposed approach.

1) Short and messy posts: The documents produced as posts in social media are significantly smaller in size mostly because of the constraints posed by social media platforms such as *Twitter*'s 140 character limit on post length. These documents may be small in size but within the short length of posts users have invented many techniques to expand the semantics of the posts. These include usage of URL shortening services (e.g, http://www.bit.ly), slang or usage of hashtags, which act like keywords starting with '#' and can be used to identify an event, phenomenon or a new concept. Hence social media documents convey rich content but in limited number of words. Hong et al. [3] has empirically shown that effectiveness of trained topics in LDA can be highly influenced by the length of the "documents". Short and noisy text of social media posts pose serious challenges to the efficacy of probabilistic topic models such as LDA which needs large documents to learn topics via word co-occurrence statistics in documents.

2) Non-uniform content generation behavior: The Pareto principle (a.k.a 80-20 rule) [4] exists almost everywhere and also applies to social networks. The volume of posts generated by users is non-uniform i.e. a small set of users generate major amount of content in social networks, for instance in [5] it has been shown that 1% of twitter users produce roughly 50% of the content on micro-blogging site. This might make contribution of niche users talking about not so popular topics in the network insignificant in the LDA model and makes the model more sensitive towards loud speakers of the network. An intuitive solution to this problem is to train a separate topic model for every user. However, to learn individual topic models for each and every user can not only become computationally expensive in large scale social network setting which comprise of millions of users, but such approach will also face cold start problem i.e. content produced by individual user is not statistically large enough to learn model parameters.

A single standard topic model trained using all social media posts as individual documents can be vastly improved. Linguistic "cleaning" could help learn a somewhat better model [6] but a number of other techniques have been proposed to improve the quality of topic models in recent years. An intuitive and popular solution to problem of short text is poolinig [3], [7], [8] i.e. merging related posts to form pseudo long texts and presenting them to LDA model as one document.

ISBN 978-952-68397-6-9 (paperback), ISBN 978-952-68397-7-6 (PDF)

The next section discuss various methods used to deal with aforementioned challenges and improve topic models for social media posts.

Social media platforms accommodate users with diverse opinions and interests, but within this diversity there exist many people who share common interests. The underlying theme of our proposed approach is based on the idea that users with similar interests tend to use limited set of popularly co-occurring words. For example, users who post about pets on Flickr have words like dog, breeds and puppies with high co-document frequency, similarly, users who talk about traveling have posts with country/city names and the word "travel" or "tour" with high co-document frequency. Many previous studies in the field of emerging topical trends and tag recommendation use this idea. In LDA where topics are defined as probabilistic distribution over words and learns from word co-occurrence statistics, pooling posts of users with similar interest therefore becomes intuitive.

We also believe that users who interact with each other (e.g. comment on each others posts) talk about a common subject or tend to be interested in topics of mutual interest and therefore use similar vocabulary. Hence, identifying communities of users with similar interests and those who interact with each other more often than others becomes useful for our purpose of preparing a training corpus for topic models by post aggregation.

In this paper, we introduce a novel document pooling scheme which is community centric. We exploit post content and network information such as user interactions, group affiliations and strong influences on users as signals to find user communities whose members can pool in their documents to train a better model.

The paper is arranged as follows. In section II we discuss existing research efforts pertaining to our work. Section III describes the dataset we have used for our experiments. Network representation, modeling user similarities and choice of community detection algorithm used in our approach of document pooling are discussed in section IV. In section V we elaborate on evaluation techniques and our choice of evaluation metrics. Experiments' parameters and results are reported in section VI. We finally conclude the paper in section VII.

II. Related work

Hong et al introduce "aggregation strategies" based on user, term etc in their work [3]. They demonstrate that better models can be trained by aggregating short messages of a user. Mehrotra et al [8] work in the same direction of aggregating posts and propose new schemes of pooling based on author, timing and hash-tags etc in order to achieve better global topic models for *Twitter*. They proove that aggregation based on the context of Hashtags yield better topics. Weng et al [7] also trained topic models on aggregated users messages.

Work has also been done to improve semantic quality of topics by making changes to the learning algorithm of LDA itself in order to maximize a quantity introduced as *coherence score* of a topic [9]. *Coherence score* is a point-wise mutual information (PMI) based score which serves as a good evaluation metric to judge the quality of topic. Maximizing it directly using generalized pólya-urn algorithm failed to make a significant difference in reducing the number of bad topics learned by a model. Nevertheless, this score has shown to highly correlate with human judgment of quality of topics and hence we use coherence score for their comparison as learned by the models.

Bindra proposes a pipeline which works in the direction of reducing cost of learning a topic model by reducing the number of documents [10]. It proposes to sample documents from users with high pagerank score in the network and shows that equally good topic models can be trained for every user in the entire network using documents generated only by the central users. They show that models trained on documents produced by a few users can be used by various other members of the social network. However they do not make any attempt at improving the quality of topic models.

Current work is built on the success of previous aggregation schemes used on social media. The new scheme does not differentiate between posts generated by users of one community and aggregate them into one long pseudo-document. This approach is generalized and can be applied to topic modeling frameworks other than LDA such as Biterm Topic Models (BTM) [11] which deals with co-occurrence sparsity by relaxing constraints on LDA and fixing corpus level topics first. If used in conjunction with our approach, BTM might learn more informative co-occurrences and corpus level topics through communities. However, in this paper, we limit our scope to learning improved topic models without modification to the LDA framework.

III. Dataset

We use Flickr datset [12] for our experiments. Flickr is a very popular photos and video sharing platform. This dataset consists of 268587 photo posts shared by 58522 users. Each post consists of hashtags given to the photo by the user. Each post in our dataset has approximately 18.36 hashtags on an average. A photograph can be part of several "groups" on the social network with different titles. Each photograph on an average is submitted to 11.77 groups. A user in flickr can add a photograph they like by other users to her own "gallery". Average number of galleries per photo is 0.62. It also consists of comments shared by users on the photographs if any.Total number of comments shared in the entire dataset are 10071439 with an average of 37.50 comments per photo.

IV. Approach

The main phases of our approach to learn improved topic models for social media posts are :

- Find groups of users who have common interests and communicate with each other more often than others.
- Prepare training corpus by pooling documents in a community as one large document and
- Finally, learn a Topic Model using corpus of aggregated documents.

In this section we discuss the work flow of our approach as shown in Fig. 1. We start by discussing how to construct different network views using post content and metadata

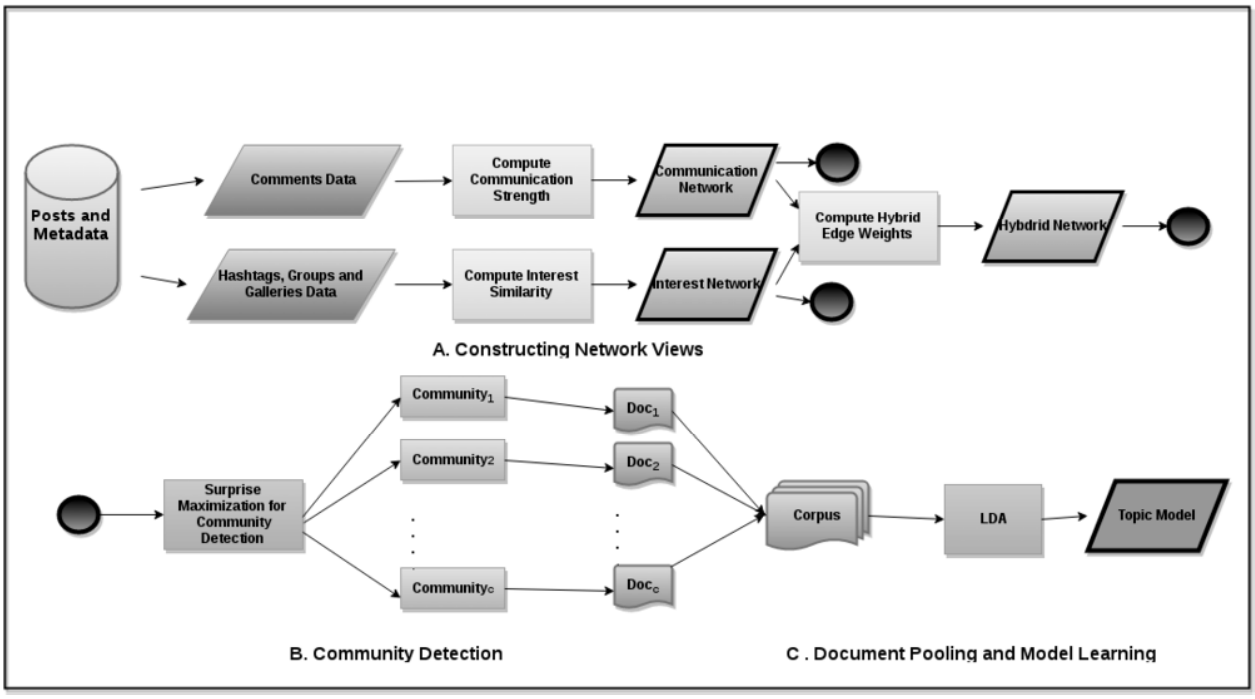

Fig. 1. Work of community based document pooling for topic model learning

available along with the posts. We also discuss in details the metrics we use to enrich the network by capturing user interest and communication strength between users. We then move on to introduce different community detection techniques and discuss our choice of community detection algorithm which is applied on network views constructed in previous phase. Finally, we discuss our community based document pooling scheme.

A. Constructing network views

The first requirement of finding a community is to form network represented by Graph, $G = (N, E, W)$. Users are added into the network as nodes *u,v* $\in$ N and edge between two users *u* and *v* with weight $w \in [0, 1]$, *e(u,v,w)* is added to the network based on the network view. Network views can be understood as multiple views of the same data. In order to find efficient communities, awareness about multiple-views of the data is important. It is closely related to multiple view clustering and is a well studied field [13]. More recently Ruan et al. [14] proposed content and topological link structure as two views to find communities in social networks. For our purpose, we use two base network views and one hybrid network view as discussed below.

1. Communication network : In social networking sites, not every neighbor is as close as a few. A user may be in more contact with very few of their neighbors. There are several features, different social networking sites provide to communicate. These include 'liking' a post, commenting on others' post, replying, messaging and many other depending on social network. For our dataset we define communication as commenting on others post. In this network, denoted by $G_{comm} = (N_{comm}, E_{comm}, W_{comm})$ edge is added to the network based on communication strength between users. Communication strength CS_{uv} between two users *u* and *v* is computed as follows:

$$CS_{uv} = \frac{C_{uv}}{C_u + C_v} \tag{1}$$

Here, C_u and C_v denotes total number of comments posted by *u* and *v* separately on anybody's post other than themselves'. And C_{uv} denotes the total number of comments posted on each others posts. For two users *u* and *v* , if $CS_{uv} > 0$ and *e(u,v,*w_{uv}*)* $\notin E_{comm}$ then we add an edge with weight $w_{uv} = CS_{uv}$, *e(u,v,*w_{uv}*)* between them.

2. Interest network : This network is denoted by $G_{interest} = (N_{interest}, E_{interest}, W_{interest})$. In this network edge, *e(u,v,*w_{uv}*)* is added on the basis of user interest.The weight of the edge between two users *u* and *v* denoted by $w_{uv} = Sim_{uv}$. Sim_{uv} quantifies user similarity on the basis of their interest similarity. It is calculated using metrics discussed below.

Content based user similarity : Content posted by user in a network is a very important signal of user interests. We measure content based similarity between two users by measuring the similarity between the hashtags used by them for their posts. Hashtags are popularly used by users in social media to indicate concept, event or a phenomenon. They serve as a good indicator of "topic" of a post. Therefore, we define edge weight between two users as similarity between sets of hashtags used by two users in their posts over time.

Hashtags are essentially words. In order to measure the similarity between sets of words, we use Word2Vec [15], [16]. Word2Vec is a word embedding technique used to compute distributed representation of words in multidimensional vector space. It is known to capture complex relationship between

words [17]. Word2vec is unsupervised in nature and we only need text documents to train a word2vec model. To train a word2vec model we treat hashtags used for one post as an individual document, we call it hashtag document. We fix the number of dimensions to 100 and use only the training dataset. We use continuous bag of word model (*CBOW*) with negative sampling. Reader is referred to [15] for details of the model. After training the word2vec model on hashtag documents, each hashtag can be represented as a 100 dimensional dense word vector. We compute content based similarity between a pair of users *u* and *v* as follows :

$$HS_{uv} = \frac{\sum_{h_i \in |H_u|} \sum_{h_j \in |H_v|} CosSim(\overrightarrow{h_i}, \overrightarrow{h_j})}{|H_u| \cdot |H_v|} \quad (2)$$

Here H_u and H_v denotes set of all the hashtags used in all the posts by users *u* and *v* respectively. $\overrightarrow{h_i}$ and $\overrightarrow{h_j}$ denotes the vector representation of hashtags obtained from the trained word2vec model. $CosSim$ is the standard cosine similarity between two vectors.

Note that Word2vec uses co-occurrence of words in immediate neighborhood and thus needs a significant context to learn meaningful embeddings. The average length of a hashtag document in our training dataset is approximately 18 words. After initial experiments and speculation we found that meaningful word embeddings could be learned from these hashtag documents, and hence we used this method for measuring content similarity. For other networks such as *Twitter* where average number of hashtags per post are significantly lower, other methods of document similarity such as cosine similarity or soft cosine similarity using WordNet [18] between *tweets* can be used.

Group affiliation based user similarity : A group in a social network serves as a niche-specific forum for it's users to share their content related to the topic of the group. Hence "groups" a user submits her posts to becomes a good indicator of user interests. A community in a network often arises from affiliation networks and group memberships also form one of the most natural affiliation network on a social media platform [19], [20]. A user can be part of various groups but some groups may be of more importance to him than others. We try to capture importance of a group *g* for a user *u* on the social media platform on the basis of their contribution to the group and popularity of the group in general. Group importance $GI_{u,g}$ is defined in *(3)*. It is based on standard tf-idf score and gives less weight to groups that are popular in the network in general.

$$GI_{u,g} = \frac{|P_{ug}|}{|P_u|} * \left(1 + log\left(\frac{N}{UF_g}\right)\right) \quad (3)$$

Here, $|P_{ug}|$ denotes total number of posts by user *u* on group *g*. $|P_u|$ denotes total number of posts by user *u*. N and UF_g denotes total number of users and number of users who have posted on *g* respectively. For a user *u* her group importance is calculated for every group and stored in a vector $\overrightarrow{GI_u}$. Similarity between two users *u* and *v* based on groups, GS_{uv} is computed as:

$$GS_{uv} = CosSim\left(\overrightarrow{GI_u}, \overrightarrow{GI_v}\right) \quad (4)$$

Influences based user similarity : In social networks some users are more influential than others. Wang et al define them as "community kernels" [21]. These are the users that generate content that other users are interested in. Various social networking sites provide different features to express interest in such users e.g. *Twitter* provides a way to express interest by following people or *re-tweeting* their *tweets*. In research community influence is expressed by citations. In Flickr it is expressed by adding posts by other users to its own "gallery".

We believe that users who express interests in content generated by same users more often than others are more similar to each other. There can be several reasons for a user to express interest in other's content depending upon the nature of the social network. In research network people can site results from other authors because they have common area of research. On *Twitter*, a user can follow or re-tweet another's post because they might keep sharing links to resources on internet which may align with her own interest. Some users in network are very popular hence users who are influenced by not so popular users are more similar to each other than users who add post by celebrity users to their galleries. Based on these ideas we try to capture influence of user *v* on another user *u* using IN_{uv} as defined below:

$$IN_{uv} = \frac{|G_{uv}|}{|G_u|} * \left(1 + log\left(\frac{N}{UF_v}\right)\right) \quad (5)$$

Here, $|G_{uv}|$ denotes total number of posts by user *v* in user *u*'s galleries, $|G_u|$ denotes total number of posts in *u*'s galleries. N, denotes total number of users and UF_v denotes number of users who have shared *v*'s posts in their respective galleries. For a user *u* influence of every other user *v* whose post she has shared is computed for her and stored in a vector $\overrightarrow{IN_u}$. Similarity between two users *u* and *v* based on influences, IS_{uv} is computed as:

$$IS_{uv} = CosSim\left(\overrightarrow{IN_u}, \overrightarrow{IN_v}\right) \quad (6)$$

for two users *u* and *v* if $Sim_{uv} > 0$ and $e(u,v,w_{uv}) \notin E_{interest}$ an edge is added between them with weight $w_{uv} = Sim_{uv}$.

$$Sim_{uv} = \alpha * HS_{uv} + \beta * GS_{uv} + (1 - \alpha - \beta) * IS_{uv} \quad (7)$$
$$\alpha, \beta \in [0, 1]$$

3. Hybrid network : In this network both aspect of communication and user interest for efficient community detection for our purpose are covered. This network is constructed by combining previously constructed *Communication network*, $G_{comm} = (N_{comm}, E_{comm}, W_{comm})$ and *Interest network*, $G_{interest} = (N_{interest}, E_{interest}, W_{interest})$ and hence called *Hybrid network*. *Hybrid network* is denoted by $G = (N_{hybrid}, E_{hybrid}, W_{hybrid})$. The new network has all the nodes and edges present in Communication network and Interest network *i.e.*

$$N_{hybrid} = N_{comm} \bigcup N_{interest} \quad (8)$$

The edge weight of hybrid network is computed as described in $Algorithm$ 1.

Algorithm 1 Hybrid Edge Weight Calculation

Input: $E_{interest}$,E_{hybrid},γ
Returns : E_{hybrid}
$E_{hybrid} \leftarrow \phi$
//Add all edges of Interest Network to Hybrid Network.
forall $e_{interest}(u, v, w_{uv}) \in E_{interest}$:
    $E_{hybrid} \leftarrow E_{hybrid} \bigcup e_{interest}$
end for
//Add all edges of Communication Network to Hybrid //Network.
forall $e_{comm}(u, v, w_{uv}) \in E_{comm}$:
    //Add edge to E_{hybrid} if edge between *u* and *v* does //not exist.
    if $e(u, v, \cdot) \notin E_{hybrid}$:
        $E_{hybrid} \leftarrow E_{hybrid} \bigcup e_{comm}$
    end if
    //Update edge weight if edge between *u* and *v* already //exists.
    else :
        $w'_{uv} = \gamma * e_{hybrid}(w_{uv}) + (1 - \gamma) * e_{comm}(w_{uv})$
        $e_{hybrid}(u, v, w_{uv}) \leftarrow e_{hybrid}(u, v, w'_{uv})$
    end else
end for
return E_{hybrid}

B. Community detection algorithms

In social networks, community detection algorithms are used to highlight its structure. Communities found using these algorithms in social networks reveal social grouping of users. Community detection in networks have numerous applications and thus it is a well studied area. There are several algorithms that have been introduced for graph partitioning and community detection. In this section we introduce basics of community detection algorithms and discuss our choice of community detection algorithm.

Clique based algorithms : A clique, C, in a network $G = (N, E)$ is a subset of the vertices, $C \subset N$, such that every two distinct nodes are adjacent. Clique based algorithms [22] aim at finding maximal cliques and label them as communities. A maximal clique is a non-extendable clique *i.e.* it can not be extended by adding one or more adjacent nodes. Clique based algorithm work fine for small networks but as the size of network increases, large cliques become hard to find. In order to solve this problem, constraints over the definition of clique are relaxed in algorithms such as k-clique and p-clique [23]. Several other algorithms such as constructing clique graphs have also been introduced [24]. However one of the biggest shortcomings of clique based algorithms is that they try to impose a structure rather than revealing natural structure in networks. Hence, they loose their applicability in social networks where there are generally too many missing links.

Modularity optimization : Modularity maximization based techniques work better at highlighting natural structure in a network. Modularity is a measure of internal density of sub-graphs as compared to external density within a network. Modularity is believed to be one of the best known measures of quality of a partition therefore maximizing modularity is at the core of many algorithms. In [25] Branden et al proved that maximizing modularity is NP complete, however algorithms such as Louvain algorithm proposed by Blondel et al [26] exist which aim at an approximate optimization using greedy approach.

Surprise optimization : Inspite of its ability to highlight natural structure, Modularity maximization techniques are well known to suffer from problem of resolution limits [23]. For our purpose we aim at finding denser niche-communities therefore we use Surprise as our quality measure. Surprise was originally proposed in [27], [28]. It is based on classical probability which describes how likely it is to find $m_{internal}$ edges in a sub-graph out of all possible ways to draw an edge in a network in fixed population of size M in m draws without replacement. More formally, surprise $S(V)$, for a partition V in a graph with m edges and n nodes is formulated as:

$$S(V) = -log \sum_{i=m_{int}}^{min(m,M_{int})} \frac{\binom{M_{int}}{i}\binom{M - M_{int}}{m - i}}{\binom{M}{m}} \quad (9)$$

Here M is the possible number of edges = $\binom{n}{2}$. And M_{int} is a partition variable which denotes total possible internal edges = $\sum_{c \in communities} \binom{n_c}{2}$ with n_c number of nodes in a community c. This formulation is hard to implement in optimization procedure due to numerical computation problems. Hence Traag et al [29] recently proposed asymptotic approximation of surprise for large graphs as :

$$S(V) \approx mD(q| < q >) \quad (10)$$

Here q represents relative number of internal edges,$< q >$ represents relative number of expected internal edges and $D(x|y)$ is the KL divergence.

$$D(x|y) = xlog(\tfrac{x}{y}) + (1 - x)log(\tfrac{1-x}{1-y}) \quad (11)$$

$$q = \tfrac{m_{int}}{m} \quad (12)$$

$$and \qquad < q >= \tfrac{M_{int}}{M} \quad (13)$$

q and $< q >$ are assumed to be fixed as the graph grows due to its asymptotic expansion for details the reader is referred to [29].

Surprise is more discriminative than modularity with an $Erdös$-$Renyi$ (ER) null model due to its use of KL divergence to quantify difference between empirical partition and the null model. For larger graphs where application demands smaller communities of almost constant size, surprise maximization is known to perform better than modularity maximization. Hence for our purpose of finding non-overlapping niche-communities in large social network, we use asymptotic surprise maximization algorithm for weighted graphs. For detailed description of the algorithm, the reader is referred to [29].

C. Document pooling and model learning

After constructing network views, surprise maximization algorithm gives us C groups of people who have common

TABLE I. COHERENCE SCORE USING TOP 10,25,50 WORDS FOR VARIOUS POOLING METHODS ACROSS NUMBER OF TOPICS

Number of Topics :	25 Topics			50 Topics			75 Topics			100 Topics		
Number of Top Words :	10 words	25 words	50 words	10 words	25 words	50 words	10 words	25 words	50 words	10 words	25 words	50 words
No Pool	-41.117	-320.449	-1541.455	-50.831	-400.141	-1867.999	-55.340	-433.954	-1979.057	-56.504	-449.596	-2068.864
User Pool	-21.749	-159.378	-808.345	-33.567	-221.376	-1049.503	-36.588	-247.267	-1176.122	-39.910	-265.120	-1205.898
Interest Pool	-16.719	-149.292	-774.166	-16.136	-161.252	-817.283	-20.854	-191.227	-939.045	-18.031	-174.651	-879.758
Communication Pool	-3.763	-75.079	-515.939	-5.332	-72.775	-521.212	-5.8954	-80.623	-523.384	-8.732	-93.719	-568.480
Hybrid Pool	**-3.6248**	**-68.262**	**-471.911**	**-4.663**	**-80.790**	**-525.343**	**-5.538**	**-80.167**	**-521.056**	**-6.038**	**-81.821**	**-539.760**

interests and communicate with each other more often than others. The last and final step in our pipeline is to pool documents in a community and use it as training corpus to learn a topic model.

As mentioned earlier a post can be part of several groups with different titles. For the purpose of training an LDA model, we treat text in each of its title, description of the post and all the hashtags used as one document. Documents are cleaned using standard NLP techniques of URL removal and stopword removal. All the documents generated by users of a community are then concatenated to make one large pseudo-document. At the end of this step we have a training corpus for LDA with C documents.

V. EVALUATION OF TOPIC MODELS

Evaluation of Topic models is an open research problem due to their unsupervised nature and varied applications. Popular metrics such as perplexity or log-likelihood over held out documents may not present a view which certainly agrees with human judgment of topics. In fact Chang et al [30] studies that topic models which perform better on held-out likelihood may infer less semantically meaningful topics. Therefore, in most studies, topic models are evaluated by presenting a sample of documents and a set of learned topics or by evaluating performance of topic models in a topical classification application if ground truth about post's topic is known. However in recent years Ruan et al [14] introduced a new PMI based evaluation metric called *Coherence Scores* which quantifies semantic quality of a topic based on document co-occurrence of top words of topics. Coherence Score are shown to be in concurrence with human experts' opinions about quality of LDA topics in the same study. Since for our work, LDA topics can also serve as description of communities' topical interests, we use *Coherence Score* described in (14) as our metric of evaluation for topic quality trained using corpus of different pooling schemes.

$$C(t;V^{(t)}) = \sum_{m=1}^{M}\sum_{l=1}^{m-1} log\frac{D(v_m^{(t)}, v_l^{(t)})+1}{D(v_l^{(t)})} \qquad (14)$$

Where $V^{(t)} = (V_1^{(1)},\ldots,V_M^{(t)})$ is a list of the M most probable words in topic t as calculated by $p(w|t)$. $D(v)$ is the document frequency of a word type *v*, and $D(v,v')$ is the co-document frequency of word types *v* and *v'*. A smoothing count of 1 is included to avoid taking the logarithm of zero.

VI. EXPERIMENTAL RESULTS

We ran our experiments for all three network views with number of topics ranging from 25 to 100. For Interest network we fix our hyper parameters α and β to $\frac{1}{3}$, giving each factor equal weightage. For Hybrid network fixing γ to $\frac{1}{2}$ gave best results after initial experimentation. We train all our LDA models using gensim's [31] implementation of variational Bayes method. All the topic models used in experimentation have symmetric Dirichlet priors. Asymmetric prior may lead to better results as suggested in [32]. However in order to reduce the effect of optimizing hyper-parameter, we fix all of them to symemetric Dirichlet priors. We fix number of passes over document to 1.

We use no-pooling and popularly used user based pooling [3] as baseline pooling techniques for comparison. We do not use Hashtag based pooling introduced in [8] since in our dataset average no of hashtags are ≈18 per post, which is way higher than number of hashtags per post in a *Twitter* dataset. Hashtag based pooling in our dataset would essentially mean sampling over every document 18 times on an average, which is equivalent to increasing the number of passes for training an LDA model.

Coherence score calculations in our experiments are done using top 10, 25 and 50 words for every topic. For each model an overall coherence score is calculated by averaging coherence score for each topic individually. Each post in the training dataset is considered as a separate document to calculate document co-occurrence frequency. Table I shows coherence score of our pooling methods across varying number of topics. Results show that popular user based pooling scheme produces more coherent topics than no pooling. This is consistent with the previous works in this area but our proposed community based pooling scheme for documents outperform user based pooling scheme and show significant improvement in the coherence score.

TABLE II. PER-WORD PERPLEXITY OVER HELDOUT DOCUMENTS FOR VARIOUS POOLING METHODS AND NUMBER OF TOPICS

PoolScheme\#topics	**25**	**50**	**75**	**100**
No Pool	-12.0598	-13.3652	-14.6806	-15.9959
User Pool	-12.946	-14.375	-16.514	-18.133
Hybrid Network Pool	**-11.991**	**-13.336**	-14.749	-16.073

Even though perplexity over held out document has shown to not always be a good predictor of human judgments of topic quality [9], [30]. Topic models are also often used to predict future text and the best known measure to evaluate quality of a topic model in this regard is perplexity [33]. Therefore for completeness of our experimentation, We use our topic models to predict held out 20% testing data. Table II shows the results of our experiments. We observe that no generalised conclusion can be drawn from the perplexity scores across topics. However we do observe that for smaller number of topics, community based model outperform other models in predicting held out documents. The disagreement between coherence scores and perplexity scores about quality of topic model is consistent with previous studies [30], [9].

Table III shows time taken to learn an LDA model. We ob-

TABLE III. LEARNING TIME(SEC) FOR VARIOUS POOLING SCHEMES

PoolScheme\#topics	25	50	75	100
No Pool	18528.11	32887.23	44545.34	54563.19
User Pool	17124.34	22205.29	32645.80	40959.96
Interest Network Pool	1878.50	3168.77	4569.35	5808.98
Communication Network Pool	2687.10	4048.10	5596.28	6907.93
Hybrid Network Pool	3080.82	4896.93	6867.55	8932.78

serve that time taken to learn an LDA model using community based pooling is much smaller than no pooling or user based pooling.

VII. CONCLUSION AND FUTURE WORK

We demonstrate that finding communities for document aggregation helps learn more coherent topic models without making any changes to LDA. We introduce weights based on influence, communication strength and group affiliations that help in finding better communities in social networks whose members can pool their documents. We also prove that creating pseudo large text documents not only produce more coherent topics but also help improve the training time of a model.

User clustering to come up with better document pooling schemes for improved topic models has not caught enough attention in the past and should be justified through more thorough theoretical analysis. In future we will analyze community detection algorithms other than Surprise maximization and try to learn optimized weights that are ideal for our purpose. It will also be interesting to analyze how other probabilistic topic models such as BTM perform in conjunction with community based pooling scheme.

REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[2] C. Reed, "Latent dirichlet allocation: Towards a deeper understanding," 2012.

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

[4] V. Pareto, *Manual of political economy: a critical and variorum edition*. OUP Oxford, 2014.

[5] S. Wu, J. M. Hofman, W. A. Mason, and D. J. Watts, "Who says what to whom on twitter," in *Proceedings of the 20th international conference on World wide web*. ACM, 2011, pp. 705–714.

[6] B. Han, P. Cook, and T. Baldwin, "Automatically constructing a normalisation dictionary for microblogs," in *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*. Association for Computational Linguistics, 2012, pp. 421–432.

[7] J. Weng, E.-P. Lim, J. Jiang, and Q. He, "Twitterrank: finding topic-sensitive influential twitterers," in *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 2010, pp. 261–270.

[8] R. Mehrotra, S. Sanner, W. Buntine, and L. Xie, "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, 2013, pp. 889–892.

[9] D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. McCallum, "Optimizing semantic coherence in topic models," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 262–272.

[10] A. Bindra, "Sociallda: Scalable topic modeling in social networks," Ph.D. dissertation, University of Washington, 2012.

[11] X. Yan, J. Guo, Y. Lan, and X. Cheng, "A biterm topic model for short texts," in *Proceedings of the 22nd international conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2013, pp. 1445–1456.

[12] J. McAuley and J. Leskovec, "Image labeling on a network: using social-network metadata for image classification," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 828–841.

[13] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *The Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2003.

[14] Y. Ruan, D. Fuhry, and S. Parthasarathy, "Efficient community detection in large networks using content and links," in *Proceedings of the 22nd international conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2013, pp. 1089–1098.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[16] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[17] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations." in *HLT-NAACL*, 2013, pp. 746–751.

[18] C. Fellbaum, *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[19] R. L. Breiger, "The duality of persons and groups," *Social forces*, vol. 53, no. 2, pp. 181–190, 1974.

[20] S. L. Feld, "The focused organization of social ties," *American journal of sociology*, pp. 1015–1035, 1981.

[21] L. Wang, T. Lou, J. Tang, and J. E. Hopcroft, "Detecting community kernels in large social networks," in *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE, 2011, pp. 784–793.

[22] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[23] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3, pp. 75–174, 2010.

[24] T. S. Evans, "Clique graphs and overlapping communities," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2010, no. 12, p. P12037, 2010.

[25] U. Brandes, D. Delling, M. Gaertler, R. Grke, M. Hoefer, Z. Nikoloski, and D. Wagner, "On modularity – np-completeness and beyond," 2006.

[26] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[27] R. Aldecoa and I. Marín, "Deciphering network community structure by surprise," *PloS one*, vol. 6, no. 9, p. e24195, 2011.

[28] V. Arnau, S. Mars, and I. Marín, "Iterative cluster analysis of protein interaction data," *Bioinformatics*, vol. 21, no. 3, pp. 364–378, 2005.

[29] V. A. Traag, R. Aldecoa, and J.-C. Delvenne, "Detecting communities using asymptotical surprise," *Physical Review E*, vol. 92, no. 2, p. 022816, 2015.

[30] J. Chang, S. Gerrish, C. Wang, J. L. Boyd-Graber, and D. M. Blei, "Reading tea leaves: How humans interpret topic models," in *Advances in neural information processing systems*, 2009, pp. 288–296.

[31] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.

[32] H. M. Wallach, D. M. Mimno, and A. McCallum, "Rethinking lda: Why priors matter," in *Advances in neural information processing systems*, 2009, pp. 1973–1981.

[33] M. Hoffman, F. R. Bach, and D. M. Blei, "Online learning for latent dirichlet allocation," in *advances in neural information processing systems*, 2010, pp. 856–864.

[34] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences*, vol. 104, no. 1, pp. 36–41, 2007.