United We Stand: Tweet Pooling for a better LDA

*Author*1[1] *Author*2[2]
(1) INSTITUTE_1, address 1
(2) INSTITUTE_2, address 2
mail-id, mail-id

ABSTRACT
Microblogging : the world of 140 characters poses serious challenges to the efficacy of topic models on short, noisy text. While Latent Dirichlet Allocation (LDA) and related models have a long history of application to news articles and academic abstracts, it has been found that standard LDA does not works well when dealing with noisy twitter data. When dealing with such noisy text, learned topics can be less coherent, less interpretable, and less useful. The goal of this work is to get better topics from twitter content without modifying the basic machinery of the standard LDA. To this effect, we present various pooling schemes to aggregate tweets for use as training data for building better LDA models. Using carefully designed experiments we propose ways of improving the quality of topics obtained, and empirically establish the competence of our work on three different kinds of datasets using three different evaluation metrics.

KEYWORDS: Topic modeling, LDA, Tweets
.

1 Introduction

As microblogging grows in popularity, services like Twitter are coming to support information gathering needs above and beyond their traditional roles as social networks. Over the past few years, Twitter has become an increasingly popular platform for Web users to communicate with each other. Because tweets are compact and fast, Twitter has become widely used to spread and share personal updates, interesting links, breaking new and spontaneous ideas. Due to the nature of microblogging, the large amount of text in Twitter may presumably contain useful information that can hardly be found in traditional information sources. The popularity of this new form of social media has attracted the attention of a lot of researchers. However, the explorations are still in an early stage and our understanding of Twitter, especially its large textual content, still remains limited.

Content analysis on Twitter poses unique challenges: posts are short (140 characters or less) with language unlike the standard written English on which many supervised models in machine learning and NLP are trained and evaluated. Tweets or status messages are short and may not carry enough contextual clues. Effectively modeling content on Twitter requires techniques that can readily adapt to the data at hand and require little supervision.

To satisfy the information needs arising from the vast Twitter data, we explore

the use of Topic Models. Topics models are probabilistic models originally developed for analyzing the semantic content of large document corpora. The ability to infer latent (or hidden) relationships between elements in data makes them more robust to handling misspellings, acronyms, terminology and other variations in the surface form of messages. Also the unsupervised nature of Topic models match well with the lack of supervision when a large collection of tweets is concerned.
While LDA and related models have a long history of application to news articles and academic abstracts, it has been found that simple LDA does not works well when dealing with noisy twitter data. Our initial investigation shows that a lot of noisy topics with incoherent topic words are thrown up when simple LDA is used on twitter datasets. In this paper we look at ways of extracting better topics from standard LDA without the need of any major modification to the basic LDA machinery. We address the problem of using standard topic models in microblogging environments by studying how the models can be trained on the dataset.

We look at various tweet-pooling schemes and compare their performances across different datasets which are characteristic of the different types of tweet-data available. Of the many pooling schemes discussed, we look at a particular kind of pooling scheme (Hashtag based pooling scheme) in detail and propose of improving the overall performance of the topic model. Evaluation of th eresultant topic model is an important issue: the unsupervised nature of topic models makes model selection difficult. We look at different evaluation metrics which evaluate the topics obtained based on different approaches including the ability of the topics obtained to reconstruct known clusters, interpretability of topics and topic coherence. More specifically, we wish to answer the following questions in the paper:

- Can we learn a topic model with better performance without any modifications to standard models?
- Do the different proposed pooling strategies perform better than unpooled tweets?
- Which pooling scheme works best across all metrics?
- Can we shed some light on how to better the best results obtained thus far? What kind of modifications are required for the same?

With a set of carefully designed experiments spanning different types of datasets and evaluation metrics which adhere to different requirements, we make the following contributions:

- Different pooling schemes perform differently across different datasets and some pooling schemes consistently outperform unpooled tweets.
- Hashtag-wise pooling leads to best results across all evaluation metrics
- We highlight how different evaluation metrics can be used to measure the performance of the different schemes.
- We present ways of assigning hashtags to tweets which further improve the performance.

Each section discusses the results in detail and insightful observations are discussed throughout the paper. The paper is organized as follows: In section 2 we look into the basics of Latent Dirichlet Allocation followed by the dataset description and evaluation metrics in section 3 and section 4 respectively. In section 5 we look at the different proposed pooling schemes and the corresponding results obtained. We next study the Hashtag based pooling scheme in detail in section 6 and present results highlighting the advantages of

assigning hashtags to tweets using different similarity metrics. Section 7 presents an overall comparison of the unpooled tweets alongside hashtag pooled and hashtag-assignment based schemes. We discuss our observations in section 8 followed by related work in section 9.

2 Topic modeling and Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA), originally introduced by Blei et al. (2003), is a generative model for text. In this model, a "topic" t is a discrete distribution over words with probability vector ϕ_t. Dirichlet priors, with concentration parameter β and base measure n, are placed over the topics $\Phi = \{\phi_1, ..., \phi_T\}$:

$$P(\Phi) = \Pi_t Dir(\phi_t; \beta n)$$

Each document, indexed by d, is assumed to have its own distribution over topics given by probabilities θ_d. The priors over $\Theta = \{\theta_1, ..., \theta_d\}$ are also Dirichlet, with concentration parameter α and base measure m:

$$P(\Theta) = \Pi_d Dir(\theta_d; \alpha m)$$

The tokens in a document $w^d = \{w_n^d\}_{n=1}^{N_d}$ are associated with topic assignments $z^d = \{z_n^d\}_{n=1}^{N_d}$, drawn i.i.d. from the document-specific topic distribution:

$$P(z_d|\theta_d) = \Pi_n \theta_{z_n^{(d)}|d}.$$

The tokens are drawn from the topics' distributions:

$$P(w_d|z_d, \Phi) = \Pi_n \phi_{w_n^{(d)}|z_n^{(d)}}.$$

A data set of documents $W = \{w(1), w(2), ..., w(D)\}$ is observed, while the underlying corresponding topic assignments $Z = \{z(1), z(2), ..., z(D)\}$ are unobserved.
Conjugacy of Dirichlets with multinomials allows the parameters to be marginalized out. For example,

$$P(z_d|\alpha m) = d\theta_d P(z_d|\theta_d) P(\theta_d|\alpha m)$$

$$= \frac{\Gamma(a)}{\Gamma(N_d+a)} \Pi_t \frac{\Gamma\left(N_{t|d}+\alpha m\right)}{\Gamma(\alpha m_t)}$$

where topic t occurs $N_{t|d}$ times in $z^{(d)}$ of length N_d.

3 Twitter Dataset Construction

Topics learned from a statistical topic model are formally a multinomial distribution over words, and are often displayed by printing the 10 most probable words in the topic. These top-10 words usually provide sufficient information to determine the subject area and interpretation of a topic, and distinguish one topic from another. However, topics learned on sparse or noisy text data are often less coherent, difficult to interpret, and not particularly

useful. Some of these noisy topics can be vaguely interpretable, but contain (in the top-10 words) one or two unrelated words – while other topics can be practically incoherent.

Focussing on noisy text of micorblogs, the primary goal of this work is to get better topics which have better coherence, interpretability and usability using standard LDA. The different pooling schemes and their proposed modifications result in different topic models, the evaluation of which is a major concern. We wish to answer questions like: Which scheme performs better on which aspects and on what kinds of data? Due to the large number of tweets in any of the twitter specific dataset, labelling of topics is not feasible. To circumvent this problem of unsupervised evaluation we carefully construct our datasets keeping the following points in mind:

- The datasets should cover the different kind of datasets possible in the microblog environment
- The method of dataset creation should help in evaluation of the different proposed schemes.

Keeping these points in mind we construct 3 different kinds of datasets catering to three different scenarios in which a tweet could be posted. We construct our dataset in such a way that it helps in evaluating the topic models obtained. An intuitive solution to the problem of unsupervised evaluation of topic models could be based on clustering approach wherein we know from before that our dataset consists of tweets from n-different clusters and then we perform topic modeling for n topics and measure how well the topic models reconstruct known clusters. We construct 3 different datasets:

1. Generic Dataset
   - Number of tweets: 359478
   - Time Duration: 11 Jan'09 - - Jan'09
   - Description: General dataset with tweets containing generic terms which represent a broader sense.
2. Specific Dataset
   - Number of tweets: 214580
   - Time Duration: 11 Jan'09 - - Jan'09
   - Description: Dataset composed of tweets which have specific terms which refer to something/someone specific.
3. Event Dataset
   - Number of tweets:
   - Time Duration:
   - Description: Event related dataset which contains tweets which were posted about some particular events. The query terms are terms which represent events.

For each of these datasets was created by querying a collection of 100 million tweets spanning 1-2 months with terms that relate to generic queries(eg. music, business, etc), specific queries(eg. Obama, McDonalds, etc) and event related queries(eg.). Tables 1-3 give the terms and the percentage tweets in the datasets which contain that term.

Term	music	business	movie	design	food	fun	health	family	sport	space
% tw	17.9	15.8	14.5	10.8	9.6	9.1	6.9	6.4	4.9	3.2

Table 1: Generic Dataset.

Term	obama	sarkozy	baseball	cricket	mcdonalds	burgerkings	apple	microso
% tw	23.2	0.4	3.5	1.8	1.5	0.5	16.3	6.8

Table 2: Events Dataset.

Term	music	business	movie	design	food	fun	health	family	sport	space
% tw										

Table 3: Specific Dataset.

It is to be noted that the 3 datasets span three different scenarios in which tweets would be posted. Evaluating our methods on each of these would give us useful insights as to which methods work well for which type of data. We next present the evaluation metrics used to compare the different topic models learnt by the different pooling schemes.

4 Evaluating Topic Models :: Metrics used

When the text being modeled is plentiful, clear and well written (e.g. large collections of abstracts from scientific literature), learned topics are usually coherent, easily understood, and fit for use in user interfaces. However, topics are not always consistently coherent, and even with relatively well written text, one can learn topics that are a mix of concepts or hard to understand. This problem is exacerbated for content that is sparse or noisy, such as tweets.
Evaluation of the different topic models based on the above mentioned features of interpretability , usability and coherence is an important issue: the unsupervised nature of topic models makes model selection difficult. For some applications there may be extrinsic tasks, such as information retrieval or document classification, for which performance can be evaluated. However, such tasks are not applicable for evaluating topics models in the microblog environment.
We evaluate our topic models based on the following 2 approaches:

1. Clustering based metrics
2. Metrics measuring Topic Coherence

We would like to measure the quality of topics found by the models. The dataset we used is the topical classification dataset containing 10 categories. Since we know the ground truth label of all the tweets in the dataset (their categories), we can measure the quality by how likely the topics agree with the true category labels. To measure how well the topics produced by LDA reconstruct known clusters, we use clustering based measures of Purity and Normalized Mutual Information (NMI). Recent work has demonstrated that it is possible to automatically measure topic coherence with near-human accuracy using a score based on pointwise mutual information (PMI).We next describe each of these measures in detail.

4.1 Purity Scores

To compute purity, each cluster is assigned to the class which is most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned documents and dividing by N. For each tweet d, we use the maximum value in topic mixture θ_d to determine its class/topic.

We interpret t_k as the set of tweets in cluster t_k and g_k as the set of tweets in category-label g_k. N is the total number of tweets; $T = \{t_1, ..., t_k\}$ is the set of k clusters and $G = \{g_1, ..., g_k\}$ is the set of k category-labels.

$$purity(T, G) = \frac{1}{N} \Sigma_k max_j |t_k \cap g_j|$$

High purity scores reflect better better cluster reconstruction, hence a topic model with high purity score is considered better.

4.2 Normalized Mutual Information

Since we know the ground truth label of all the tweets in the dataset, i.e., their categories, we can measure the quality by how likely the topics agree with the true category labels.

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

where $I(T, G)$ is Mutual Information and $H(T)$ gives the entropy. The corresponding values are:

$$I(T, G) = \Sigma_k \Sigma_j \frac{|t_k \cap g_j|}{N} log \frac{|t_k \cap g_j|}{|t_k||g_j|}$$

NMI is always a number between 0 and 1. NMI may achieve 1 if the clustering results can exactly match the category labels while 0 if the two sets are independent. For each tweet d, we use the maximum value in topic mixture θ_d to determine its cluster. After this mapping process, we compute NMI scores with the labels.
High purity is easy to achieve when the number of clusters is large. The normalization by the denominator in the NMI fixes this problem as entropy tends to increase with the number of clusters.

4.3 Pointwise Mutual information :: Topic Coherence

One of the goals of our work is to get topics which are more coherent. Topic coherence – meaning semantic coherence – is a human judged quality that depends on the semantics of the words, and cannot be measured by model-based statistical measures that treat the words as exchangeable tokens. Recent work has demonstrated that it is possible to automatically measure topic coherence with near-human accuracy using a score based on pointwise mutual information (PMI).

$$PMIScore(w) = Mean(PMI(w_i, w_j))i, j \in \{1.....10\}$$

A key aspect of this score is that it uses external data. We eliminate the need of this external data source by using our own dataset to calculate the probabilities used in this score.

5 Tweet Pooling

5.1 Pooling Schemes

As microblogging grows in popularity, services like Twitter are coming to support information gathering needs above and beyond their traditional roles as social networks. Studying the characteristics of content in the messages becomes important for a number of tasks, such as breaking news detection, personalized message recommendation, friends recommendation, sentiment analysis and others.
Microblog messages differ from conventional text. They feature many unique symbols like mentions, hashtags and urls, and the popular use of colloquial words and Internet slang. Message quality varies greatly, from newswire-like utterances to babble (e.g. O o haha wow). In terms of text processing, there are significant research challenges.
While many researchers wish to use standard text mining tools to understand messages on Twitter, the restricted length of those messages and the differences with standard text discussed above prevents them from being employed to their full potential. One such tool is LDA for topic modeling.

When the text being modelled is well structured, clear and well written (e.g. large collections of news articles), learned topics are usually coherent, easily understood, and fit for use in user interfaces. However, when dealing with small collections or noisy text (microblog content), learned topics can be less coherent, less interpretable, and less useful. When used without any preprocessing, LDA performs miserably on collection of tweets. Figure 1 shows an example of the kind of topics obtained when tweets are fed as is to a standard LDA. The topics that emerge are very noisy with incoherent topic words which are less interpretable and thus less useful.

The goal of this paper is to get better topics from twitter content without modifying the basic machinery of standard LDA. To this effect we present various pooling schemes to aggregate tweets together for use as training data for building better LDA models. The motivation behind tweet pooling is that individual tweets are very short(<=140 characters) and hence treating each tweet as a document does not works well for topic modeling. We next describe few tweet pooling schemes and compare the resulting models against the one obtained with unpooled tweets.

1. **Basic scheme: Unpooled Tweets**
   The default way of training models involves treating each tweet as a single document and training LDA on all tweets. This serves as our baseline and we compare results of other pooling schemes against this Unpooled scheme. For each tweet d, using the trained model, we use the maximum value in topic mixture θ_d to determine its class/topic and use theresult to calculate the Purity and NMI scores.

2. **Burst Score wise Pooling**
   A *trend* on Twitter (sometimes referred to as a trending topic) consists of one or more terms and a time period, such that the volume of messages posted for the terms in the time period exceeds some expected level of activity. In order to identify trends in Twitter posts, "bursts" of interest and attention can be detected in the data. We run a simple burst detection algorithm to detect such trending topics and aggregate

tweets containing those terms having high burst scores.
Specifically, to identify terms that appear more frequently than expected, we will assign a score to terms according to their deviation from an expected frequency. Assume that M is the set of all messages in our Tweets dataset, R is a set of one or more terms to which we wish to assign a score, and d represents a day of the total n days. We then define $M(R, d)$ as the set of every Twitter message in M such that (1) the message contains all the terms in R and (2) the message was posted during day d. With this information, we can compare the volume in a specific day to the other days. Let

$$Mean(R) = \frac{\Sigma_d M(R,d)}{n}$$

Correspondingly, $SD(R)$ is the standard deviation of the number of messages with the terms in R posted over all the days. The *burst-score* is defined as:

$$burst - score(R, d) = \frac{|M(R,d) - Mean(r)|}{SD(R)}$$

Let us denote the terms having burst-scores greater than 5 (empirically calculated value) as *burst-term*. We create the burst score-wise pooling scheme using these burst-terms as follows:

- For each burst-term, aggregate tweets which contain this term.
- Train LDA on this aggregation of tweets
- Use the train model to infer a topic mixture for each of the individual tweets.

This scheme is henceforth referred to as Burst Score-wise pooling.

3. **Author-wise Pooling**
   Another way of tweet pooling could be to aggregate tweets posted by a particular author as a single document and repeat this for all the authors. We train LDA on aggregated author profiles, each of which combines all tweets posted by the same author. Using this model we infer a topic mixture of individual tweets.

4. **Temporal Pooling**
   The fourth scheme, known as Temporal Pooling, is utilizes the temporal information of the tweets. It is noted that whenever a major event occurs a large number of users start tweeting about the event. Temporal pooling of tweets might help us in extracting useful information from the LDA topics. We train the LDA on an aggregate of tweets posted within the same hour and use this model to infer a topic mixture for each of the individual tweets.

5. **Hashtag-wise Pooling**
   A Twitter *hashtag* is a string of characters preceded by the hash (#) character. In many cases hashtags can be viewed as topical markers, an indication to the context of the tweet or as the core idea expressedin the tweet, therefore hashtags are adopted by other users that contribute similar content or express a related idea. A few examples of the use of hashtags are: "ask GAGA anything using the tag #GoogleGoesGaga for her interview! RT so every monster learns about it!! " referring to an exclusive interview for Google by Lady Gaga (singer), "Whoever said 'youth is wasted on the young' must be eating his words right now. #March15 #Jan25 #Feb14 ", referring to the protest movements in the Arab world.
   For the hashtag based pooling scheme, for each hashtag we aggregate tweets containing this hashtag and train LDA on this collection. Each hashtag pooled tweet collection

thus represents a document. We notice that this tweet pooling scheme outperforms the rest. We discuss hashtag based pooling in detail in a later section.

5.2 Document Characteristics for different pooling schemes

Here we look at the document characteristics of the documents in the different pooling schemes for all three datasets. The characteristics like the number of documents affect LDA directly and hence it will be interesting to look at what the training data comprises of. Table 3 presents the required statistics.

Pooling Scheme	#of docs			Avg # of words/doc			Max # of words/doc		
	DS1	DS2	DS3	DS1	DS2	DS3	DS1	DS2	DS3
Unpooled	359478	214580	-	10.20	10.94	-	65	79	-
Authorwise	208300	118133	-	17.60	20.47	-	4893	3586	-
Hourly	465	464	-	8493.4	5387.5	-	20144	18869	-
Burst Score	10	20	-	10	20	-	10	20	-
Hashtag	8535	7029	-	70.44	187.28	-	61918	420249	-

Table 4: Document Characteristics for different schemes

5.3 Initial Results

In this section we discuss the results of the experimental evaluation of the tweet pooling schemes discussed previously. The datasets used were described in section 3 while the evaluation metrics used were described in section 4. For each of the three datasets (viz. Generic, Specific and Events) we present the Purity scores, the NMI scores and the PMI scores in Table 4, 5 and 6 respectively.

Pooling Scheme	Unpooled	Author	Hourly	Burstwise	Hashtag
Generic	0.49	0.55	0.0.45	0.0.42	0.0.53
Specific	0.64	0.62	0.61	0.60	0.68
Events	-	-	-	-	-

Table 5: Purity Scores for different datasets

Pooling Scheme	Unpooled	Author	Hourly	Burstwise	Hashtag
Generic	0.28	0.24	0.07	0.18	0.28
Specific	0.22	0.17	0.09	0.16	0.23
Events	-	-	-	-	-

Table 6: NMI Scores for different datasets

Pooling Scheme	Unpooled	Author	Hourly	Burstwise	Hashtag
Generic	-1.27	0.21	-1.31	0.48	0.78
Specific	0.47	0.79	0.87	0.74	1.43
Events					

Table 7: PMI Scores for different datasets

Based on these results we conclude that Hashtag wise pooling scheme performs better than unpooled scheme as well as other pooling schemes. An obvious question to ask is: Can we do better? In the next section we look into Hashtag-wise pooling in detail and devise methods which further improve the results and provide better topics.

6 General Study of Hashtags & Hashtag Pooling

Hashtag based tweet pooling outperforms other pooling schemes and unpooled tweets in terms of both: reconstructing known clusters as well as extracting coherent topic words. In this section we look into hashtags and hashtag based pooling in detail and analyse the prominence of hashtags in our datasets and look at ways to further improve the results.

6.1 Do all tweets have hashtags?

Among all tweet pooling schemes Hashtag based pooling gives the best results in terms of purity scores, NMI and PMI values. Few obvious questions to ask at this stage include: How common are the hashtags? Do all tweets have hashtags? Table 9 presents few statistics on the percentage of tweets which do not have any hashtag in the 3 datasets.

In this section we look at the number of tweets which do not have any hashtag and posit problems which arise due to this.

Dataset	% tweets having hashtags
Generic	22.3%
Specific	9.4%
Event	15%

Table 9: % tweets haing hashtags

As is evident from the figures a large number of tweets do not contain any hashtag, with the percentages varying from 77.7% to a surprising 91.6%. This suggests that a large portion of the training data does not participates in the hashtag pooling scheme. Since a large portion of the data is getting ignored we need to figure out ways of incorporating this data while training LDA models. We next address this issue and present a brute force way of doing so.

6.2 Incorporating Other Tweets :: Brute Force Way

As discussed in the previous section, atleast 77% of tweets do not have any hashtags. One ways of incorporating this left out pat of the dataset could be to include the entire remaining portion as is alongside hashtag pooled collection of tweets. Table 10 shows the results on different datasets when the entire remaining part of tweets which do not have any hashtag are included alongside hashtag pooled tweets and the percentage improvement.

Metric	Generic		Specific		Events	
	Full	% improvement	Full	% improvement	Full	% improvement
Purity	0.60	+13.2	0.69	+1.4	-	-
NMI	0.29	+3.5	0.23	+0.1	-	-
PMI	0.42	-46.1	0.59	-58	-	-

Table 10: Results of incorporating other tweets on different datasets

The results in the above table suggests that this technique harms the topic coherence in a very bad manner(lower PMI scores) but improves cluster reconstruction(higher purity scores). We need to look at ways in which we could improve cluster reconstruction without adversely affecting the topic coherence. In the next section we present a way of doing so and show that our proposed method performs better than the results so far.

6.3 Assigning Hashtags

The brute force way of incorporating tweets without hashtags terribly degrades the PMI scores alongside improving the purity scores. In this subsection we wish to look into another ways of incorporating these tweets so as to balance out improvements in cluster reconstruction and degradation of topic coherence.Since a large number of tweets do not have hashtags we propose that assigning hashtags to some of those tweets help in improving overall results. We discuss here an algorithm which assigns a hashtag to some of the tweets and hence increases the number of tweets having atleast 1 hashtag.

6.3.1 Algorithm

Our aim here is to assign hashtags to tweets which have none. Since this step is done after hashtag based pooling of tweets, for each hashtag we have a collection of tweets each of which contain this particular hashtag. We make use of this collection to compare each tweet with each hashtag's collection and compute the similarity scores. If the similarity score crosses a certain threshold(computed empirically) we assign the hashtag to this tweet and add this tweet to the pool of tweets assigned to this hashtag. The similarity metric used here is the tf-cosine similarity. Table 11 describes this algorithm in detail.

6.3.2 Hashtag Assignment Results

Table 12 presents the results of hashtag assignment based on the 3 metrics for different threshold varying from 0.5 to 0.9. We notice that as the threshold increases the corresponding value of PMI scores increases and purity scores decreases which is intuitive: on increasing the threshold lesser number of tweets get assigned a hashtag and hence cluster reconstruction suffers while the topic coherence is improved because a tweet is assigned a hashtag only if its highly similar to the tweet collection of that hashtag(threshold = 0.9).

The results obtained via hashtag assignment are better than those of simple hashtag based pooling as well as unpooled scheme. These results were obtained on using tf-cosine similarity metric to compute the similarity between a tweet and the tweet collection of the hashtag in consideration. We next look at some other similarity metrics and see if we could further improve our results.

6.3.3 Other Similarity Metrics

In this section we discuss in detail the effects of using different similarity metrics for hashtag assignment and later provide a new metric which gives improved results.

Tf-IAF

Description

6.3.4 New Similarity Metric

Description

6.3.5 Results of new Similarity Metrics

Here we discuss results using the new similarity metrics

7 Overall Comparison

The goal of this work was to get better topics which have better coherence, interpretability and usability without modifying the basic machinery of standard LDA. The default way of using tweets to train LDA was to treat each tweet as a single document (referred in this work as the Unpooled scheme). Simple hashtag based pooling outperformed unpooled scheme and other pooling schemes while hashtag assignment further improved results. Table 14 provides an overall comparison of the different schemes: Unpooled vs simple hashtag vs hashtag assignment on the three datasets.

Algorithm 1: HASHTAG ASSIGNMENT

input : H_T : set of tweets having hashtags (N in total)
N_T : set of tweets with no hashtag (M in total)
T : set of hashtags
c = threshold on the similarity score
output: hashtag h (in T) for tweet $n_t \in N_T$ if score(n_t,h) > c

```
begin
    // –Pooling Step–
    foreach hashtagh ∈ T do
        collect all tweets containing hashtag h
    max_score = 0
    // –Assignment Step–
    foreach tweetn_t ∈ N_T do
        foreach hashtagh ∈ T do
            calculate similarity score sim_score b/w n_t and tweet pool of hashtag h
            if (sim_score > max_score)
            then temp_assigned_tag = h ; max_score = sim_score

        if (max_score > c)
        then assigned_tag = temp_assigned_tag

end
```

Threshold	Purity			NMI Score			PMI score		
	DS1	DS2	DS3	DS1	DS2	DS3	DS1	DS2	DS3
0.5	0.59	**0.72**	-	**0.356**	**0.242**	-	0.70	1.10	-
0.6	0.59	0.68	-	0.334	0.221	-	0.59	1.11	-
0.7	**0.62**	0.70	-	0.31	0.222	-	0.66	1.12	-
0.8	0.55	0.69	-	0.295	0.225	-	0.72	1.16	-
0.9	0.53	0.69	-	0.28	0.227	-	**0.82**	**1.21**	-

Table 12: Hashtag Assignment Results

Threshold	Purity			NMI Score			PMI score		
	DS1	DS2	DS3	DS1	DS2	DS3	DS1	DS2	DS3
0.5	10	20	-	10	20	-	10	20	-
0.6	10	20	-	10	20	-	10	20	-
0.7	10	20	-	10	20	-	10	20	-
0.8	10	20	-	10	20	-	10	20	-
0.9	10	20	-	10	20	-	10	20	-

Table 13: Hashtag Assignment Results based on New Similarity Metric

Pooling Scheme	Purity			NMI Scores			PMI Scores		
	DS1	DS2	DS3	DS1	DS2	DS3	DS1	DS2	DS3
Unpooled	0.49	0.64	-	0.29	0.22	-	-1.27	0.47	-
Basic Hashtagwise	0.53	0.68	-	0.28	0.23	-	0.78	**1.43**	-
Tag-Assignment	**0.62**	**0.72**	-	**0.36**	**0.24**	-	**0.82**	1.21	-

Table 14: Overall comparison of improvement

8 Observations

Herewe list down pointwise our observations, the affect of various settings and our main findings/insights.

9 Related Work

Topic modeling is gaining increasing attention in different text mining communities. Latent Dirichlet Allocation (LDA) [1] is becoming a standard tool in topic modeling. LDA has been extended in a variety of interesting ways, and in particular for social networks and social media, a number of extensions to LDA have been proposed. For example, Newman et al. [2] proposed two methods to regularize the learning of topic models aimed at short text snippets.

Among the social networks, several characteristics of the data available in microblogs make them appealing for applications which fall into three broad categories: event detection [3], trend identification [4], and social group identification [5]. The interested reader is directed to the work of Liao et al[9] which discusses the opportunities and challenges related to mining microblogs.

Our work is quite different from many pioneering studies on Twitter and topic modeling

because we try to study how we could get better topics using tweets with minimalistic efforts. Prior work on topic modeling for tweets includes the work of Ramage et al[6] which presents a scalable implementation of a partially supervised learning model (Labeled LDA) that maps the content of the Twitter feed into dimensions and characterize users and tweets using this model. Zhao et al [7] empirically compare the content of Twitter with a traditional news medium, New York Times, using unsupervised topic modeling. Hong et al [8] use the topic modeling approach for predicting popular Twitter messages and classifying Twitter users and corresponding messages into topical categories.
Evaluating topic models has continued to be an active research topic with the aim of automatically evaluating topic models. Newman et l [10] introduce the task of topic coherence evaluation, whereby a set of words, as generated by a topic model, is rated for coherence or interpretability. Wallach [11] present evaluation methods based on the probability of held-out documents given a trained model.

10 Conclusion

Acknowledgments

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References