Improving Topic Models for Microblogs : 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 generally messy Twitter data. When dealing with such 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 effectiveness of our work on three different kinds of datasets using three different evaluation metrics.

Keywords: Topic modeling, LDA, Tweets
.

1 Introduction

In the general area of information retrieval or information access, the *undirected informational task* [Rose and Levinson, WWW 2004] is one where people seek to better understand the information available to them in a particular area. This is a form of *information discovery* that techniques such as multidocument summarisation [Radev, Hovy, and McKeown, Computational linguistics, 2002] and topic modelling hope to address.

Probabilistic topic models are a class of Bayesian latent variable model, originally developed for analyzing the semantic content of large document corpora. Topic models uncover the salient patterns of a collection under the mixed-membership assumption: each data instance can exhibit multiple patterns to different extent. When analysing text, these patterns are represented as distributions over words and often place high probability on semantically meaningful sets called "topics". With the increasing availability of large, heterogeneous data collections, topic models have been adapted to model document genres as diverse as news articles [12], blogs, academic abstracts [13], and encyclopaedia entries.

To satisfy the undirected informational task arising from the vast data from Twitter, we are exploring the use of topic models. However, Twitter content poses unique challenges different to much of standard NLP content:

- posts are short (140 characters or less),
- mixed with some contextual clues such as URLs, tags, and twitter names, and
- using informal language unlike the standard written English with misspelling, acronyms and nonstandard abbreviations.

on which many Tweets or status messages are thus short and linguistically messy making it difficult to get contextual and semantic clues. Effectively modeling content on Twitter requires techniques that can readily adapt to this kind of data and require little supervision.

It has been found that simple LDA [1] does not works well when dealing with the messy Twitter data[7]. Topics learned from LDA are formally a multinomial distribution over words, and by convention the top-10 words are used to identify the subject area or give an interpretation of a topic, thereby distinguishing one topic from another. The application of LDA on Twitter data produces messy topics with incoherent topic words which are difficult to interpret. While few topics are vaguely interpretable but contain unrelated words in the top-10 word set.

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 various tweet pooling schemes and compare their performance against the default way of training LDA: each tweet representing a document. The bad performance of LDA on Twitter data can be attributed to the fact that in an unpooled setting each document is small (140 characters) as well as written in improper grammar, hence an individual document does not contains enough content which might help LDA to discover latent semantics in a document.

An intuitive solution to this problem is tweet pooling: merging related tweets together and presenting them as a single document to learn the LDA model. In this work we look at various tweet-pooling schemes and compare their performances across different datasets. The datasets are constructed in a way that they are representative of the diverse collections of content possible in the microblog environment. We observe that of the different pooling schemes discussed, hashtag based tweet pooling scheme outperforms the others across all datasets. We further look into Hashtag based tweet pooling scheme and suggest ways of assigning hashtags to tweets which do not have any hashtag, which further improves the quality of topics obtained. For the task of hashtag assignment, we look at various similarity metrics and compare the quality of topics obtained.

Evaluation of the resultant topic model is challenging because of the unsupervised nature of topic models. 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. We wish to answer the following questions in the paper:

- Can we learn an LDA topic model with better performance without any modification to the core LDA algorithm?
- Do the different proposed pooling strategies perform better than unpooled tweets?
- Which pooling scheme work best for which metric and why?

- What further improvements can be made to various pooling schemes so as to obtain topics which are coherent and interpretable.

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

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

In the following sections we describe the proposed tweet pooling schemes and demonstrate that tweet pooling improves the quality of the topics obtained by a big margin (of the order of ten for one of the evaluation metrics). The Hashtag-based pooling scheme and tag assignment proposed works better than all other schemes across almost all datasets and metrics.

2 Topic Modelling and Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA), originally introduced by Blei et al.[1], 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 $w^d = \{w_n^d\}_{n=1}^{N_d}$ (where N_d is the number of tokens in the document 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.

We use the Mallet[4] implementation of LDA for performing our experiments.

3 Twitter Dataset Construction

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.

Focusing on the messy text of microblogs, the primary goal of this work is to get topics which have better coherence and interpretability using standard LDA. We discuss the following five tweet pooling schemes, which are described in detail in a later section:

- Unpooled
- Author wise pooling
- Temporal Pooling
- Burst score wise pooling
- Hashtag based pooling

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 (360000) in any of the twitter specific datasets, manual 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 diverse collections of content possible in the microblog environment
- The method of dataset creation should help in evaluation of the topic model obtained using the different tweet pooling schemes.

Based on the above mentioned points we construct three datasets which we believe are representative of the diverse collections of content found in Twitter data. From a large collection of tweets we choose single/double word queries to search the tweet collection and each resulting set of tweets was labelled by the query. Since the number of queries (equivalently the number of clusters) is known beforehand, we could use this knowledge to evaluate how well the topics output by LDA match with known clusters. A brief description of the three datasets is as follows:

1. Generic Dataset
   - Number of tweets: 359478
   - Time Duration: 11 Jan'09 - 30 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 - 30 Jan'09
   - Description: Dataset composed of tweets which have specific terms which refer to named entity topics.
3. Event Dataset
   - Number of tweets: 207128
   - Time Duration: 1 June 2009 - 30 June 2009
   - Description: Event related dataset which contains tweets which were posted about some particular events. The query terms are terms which represent events.

Each of these datasets was created by querying a collection of 100 million tweets spanning 2 months (January'09 & June'09) with terms that relate to generic queries (broad topic words like music, business, etc), specific queries(named entity topics like Obama, McDonalds, etc) and event related queries(actual events in that timeframe like recession, Flight 447, Iran elections, etc). Tables 1,2 and 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	microsoft	united statess	france
% tw	23.2	0.4	3.5	1.8	1.5	0.5	16.3	6.8	40.7	4.9

Table 2: Events Dataset.

Term	Flight 447	Jackson	Lakers	attack	scandal	swine flu	recession	conference	T20	Iran election
% tw	0.9	13.9	13.8	13.8	4.1	13.8	12.3	14.1	4.4	8.6

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 modelled 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, but the unsupervised nature of topic models makes this 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 *undirected informational task*.

We evaluate our topic models based on the following two approaches:

1. **Clustering based metrics**
   We would like to measure the quality of topics found by the models. The dataset we used is a 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).

2. **Metrics measuring topic coherence**
   It is desirable that the learnt topics are coherent and interpretable. Recent work[10] has demonstrated that it is possible to automatically measure topic coherence with near-human accuracy using a score based on pointwise mutual information (PMI). PMI scores serve as our metric to measure coherence of the topics output by different tweet-pooling schemes.

We next describe each of these measures in detail.

4.1 Purity Scores

To compute purity[15], 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. m 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 (e.g. Obama, Microsoft).

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

where

$$t_k = \{d \mid argmax_{t^*} \theta_d^{t^*} = t\}$$

As the number of correctly assigned tweets increases for each cluster, the overall purity score increases. hence high purity scores reflect 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:

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

$$H(T) = -\Sigma_k \frac{|t_k|}{N} log \frac{|t_k|}{N}$$

NMI[15] is always a number between 0 and 1. NMI score will be 1 if the clustering results 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

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[12] 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. Newman et al[16] used the Wikipedia corpus as their 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

The goal of this paper is to get better topics from Twitter content without modifying the basic machinery of standard LDA. 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. 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($\leq$ 140 characters) and hence treating each tweet as a document does not works well for topic modelling. Aggregating tweets which are similar in some sense (semantically, temporally etc) enriches the content present in a single document from which the LDA can learn better topic model. We next describe four different 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 y represents a day of the total z days. We then define $M(R, y)$ 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 y. With this information, we can compare the volume in a specific day to the other days. Let
$$Mean(R) = \frac{\Sigma_d M(R,y)}{z}$$
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:
$$burst\_score(R, y) = \frac{|M(R,y) - Mean(r)|}{SD(R)}$$
Let us denote the terms having burst-scores greater than 5 as *burst-term*. For each burst-term, aggregate tweets which contain this term and train LDA on this aggregation of tweets. We then use the trained 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 expressed in 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. If any tweet has more than one hashtag, this tweet gets added to the tweet-pool of each of those hashtags. This results in tweets being repeated across documents. 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 the 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 consists of. Table 3 presents the required statistics.

Pooling Scheme	#of docs			Avg # of words/doc			Max # of words/doc		
	Generic	Specific	Events	Generic	Specific	Events	Generic	Specific	Events
Unpooled	359478	214580	207128	10.2	10.9	9.7	35	49	32
Burst Score	7658	7436	5434	76.5	154.2	71.6	61918	420249	57794
Hourly	465	464	463	8493.4	5387.5	2422	20144	18869	38893
Authorwise	208300	118133	67387	17.6	20.4	15.4	4893	3586	2775
Hashtag	8535	7029	4099	70.4	187.2	78.4	61918	420249	57794

Table 4: Document Characteristics for different schemes

The statistics presented above highlight the differences in the characteristics of the documents on which LDA models have been trained. The number of documents decreases as we move from Unpooled scheme to Authorwise and Hashtagwise pooling scheme, while the corresponding size of the documents in each case increases. On an average the document size increases by a factor of 7 in hashtag based pooling when compared against unpooled or authorwise pooling schemes. Thus each document in hashtag based pooling contains more content from which LDA could possibly extract latent semantics. On the other extreme lies the temporal pooling with very less number of documents and hence each document of a much larger size. Such large documents might impact the topic model in an unpleasant manner. These statistics highlights that hashtag based pooling scheme lies mid-way between both the extremes (small documents in unpooled tweets vs large documents in temporal pooling) and hence suggests that hashtag based pooling should perform optimally in comparison to other 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.

Based on these results we conclude that Hashtag wise pooling scheme performs better than

Pooling Scheme	Unpooled	Author	Hourly	Burstwise	Hashtag
Generic	0.49	**0.54**	0.45	0.42	**0.54**
Specific	0.64	0.62	0.61	0.60	**0.68**
Events	0.69	0.70	0.61	0.64	**0.71**

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	0.39	0.41	0.32	0.33	**0.42**

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	0.47	0.51	0.22	0.58	**1.07**

Table 7: PMI Scores for different datasets

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. There are two 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	19.5%

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, at least 77% of tweets do not have any hashtags. One ways of incorporating this left out part 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%	0.68	-1.5%
NMI	0.29	+3.5%	0.23	+0.1%	0.40	+2.5%
PMI	0.42	-46.1%	0.59	-58%	1.2	+155%

Table 10: Results of incorporating other tweets on different datasets. % improvements are measured in comparison with simple hashtag pooling results.

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 at least 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(burst score 5) 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. To avoid the bias caused by different document lengths, a common way to compute the similarity of two documents is using the cosine similarity measure. The inner product of the two vectors (sum of the pairwise multiplied elements) is divided by the product of their vector lengths. This has the effect that the vectors are normalized to unit length and only the angle, more precisely the cosine of the angle, between the vectors

accounts for their similarity. Documents not sharing a single word get assigned a similarity value of zero because of the orthogonality of their vectors while documents sharing a similar vocabulary get higher values (up to one in the case of identical documents).

$$sim(d1, d2) = \frac{d1.d2}{||d1||||d2||}$$

where d1 is the tweet in consideration to which we intend to assign a hashtag and d2 is the collection of tweets for some hashtag. We iterate through all the hashtag for different d2 vectors and assign the hashtag which maximizes this similarity score. We use the term frequency of the words in the tweet to represent the tweet as a tf vector. Algorithm 1 describes the same algorithm in detail.

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
```

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

Threshold	Purity			NMI Score			PMI score		
	Generic	Specific	Events	Generic	Specific	Events	Generic	Specific	Events
0.5	0.59	**0.72**	**0.75**	**0.356**	**0.242**	**0.442**	0.70	1.10	0.92
0.6	0.59	0.68	0.73	0.334	0.221	0.437	0.59	1.11	0.96
0.7	**0.62**	0.70	0.72	0.31	0.222	0.431	0.66	1.12	0.98
0.8	0.55	0.69	0.72	0.295	0.225	0.429	0.72	1.16	1.0
0.9	0.53	0.69	0.71	0.28	0.227	0.42	**0.82**	**1.21**	**1.05**

Table 12: Hashtag Assignment Results

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

The results presented in Table 12 use the tf-cosine similarity between tweet and tweet-collection of a hashtag to assign a hashtag to the tweet in consideration. We next answer the question of how other similarity metrics affect the results of hashtag assignments. We look at the following six metrics to be used to represent each document while calculating the cosine similarity scores:

1. **Term Frequency (TF)**: Already discussed in the previous section.
2. **Inverse Document Frequency (IDF)**: a measure of whether the term is common or rare across all documents. It is obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient. Thus the idf of a rare term is high, whereas the idf of a frequent term is likely to be low.
3. **Inverse Author Frequency (IAF)**: Each term used in a tweet has a relationship to the tweet's author. The relationship of terms and authors could be harnessed using the inverse author frequency (IAF) [17] which is computed as follows:

   $$iaf_i = log\frac{N}{n_i}$$

   where N : # of total authors and n_i : # of authors who use term i
4. **TF-IDF**: the product of the term frequency and inverse document frequency of each term.
5. **TF-IDF-IAF**: the product of the term frequency, inverse document frequency and the inverse author frequency of each term.

6.3.4 Results of other similarity metrics

Figure 2 presents the results obtained for the three evaluation metrics on the three datasets for the different similarity metrics discussed in the previous section. It is interesting to note that the simplest of all: term-frequency based cosine similarity generally performs better than others. Inverse Author Frequency (IAF) gives almost comparable results when

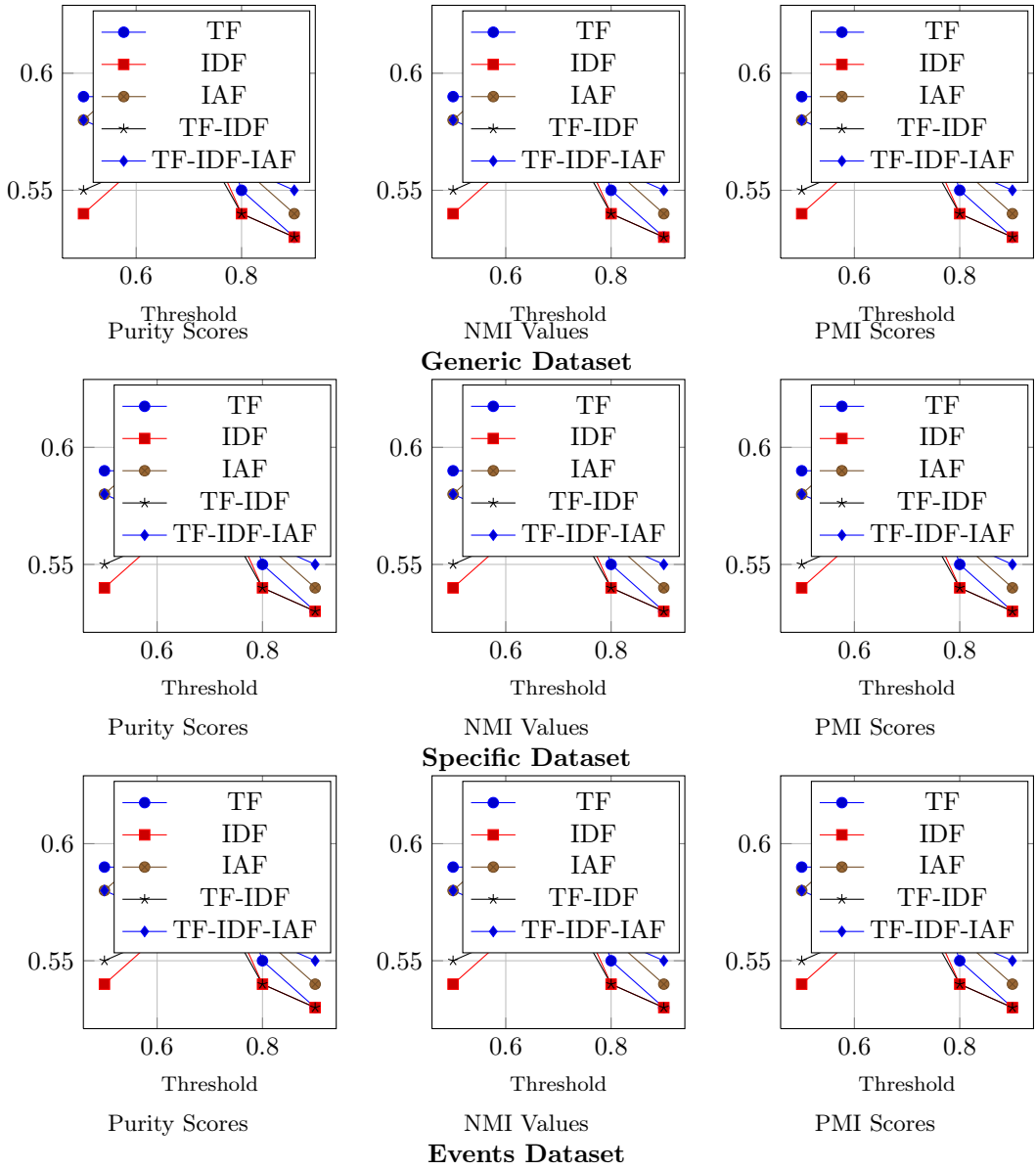

Figure 1: Comparison of the peformance of the different similarity metrics on different datasets.

evaluating topic coherence using the PMI scores. When the task at hand is reconstruction of known clusters one should prefer using TF based cosine similarity to assign hashtags while both TF and IAF work well when one is interested in obtaining coherent topics.

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 13 provides an overall comparison of the different schemes: Unpooled vs simple hashtag vs hashtag assignment on the three datasets.

Pooling Scheme	Purity			NMI Scores			PMI Scores		
	Generic	Specific	Events	Generic	Specific	Events	Generic	Specific	Events
Unpooled	0.49	0.64	0.69	0.29	0.22	0.39	-1.27	0.47	0.47
Basic Hashtagwise	0.53	0.68	0.71	0.28	0.23	0.42	0.78	**1.43**	**1.07**
Tag-Assignment	**0.62**	**0.72**	**0.75**	**0.36**	**0.24**	**0.44**	**0.82**	1.21	1.05

Table 14: Overall comparison of improvement

8 Observations

The work described in this paper presents a way of aggregating tweets in order to improve performance of topic models in terms of quality of topics obtained measures by the ability to reconstruct clusters and topic coherence. The initial results presented in Table 5,6 and 7 suggest that hashtag based pooling outperforms all other pooling strategies including the default way of training topic models on Twitter data(unpooled).

Since a major portion of Twitter data does not contains hashtags we looked at ways of assigning hashtags to tweets. Insights from Hashtag Assignment results (Table 12) suggest that when the main aim is to use the topics obtained to extract different events mentioned in the Twitter data one should use hashtag assignment with a relaxed threshold(0.5). The high values of Purity scores and NMI values for low threshold support this claim.

When the goal is to obtain interesting topics with topic words pertaining to the same common theme (coherent topic words), hashtag assignment with strict constraints(threshold 0.9) works well. The PMI scores in Table 12 highlight that topic coherence increases as we move down the column. Thus the threshold of 0.9 gives most coherent topics.

Overall, results shown in Table 13 compare unpooled scheme with simple hashtag pooling and hashtag assignment schemes. The best Purity and NMI scores are obtained by hashtag assignment while even the simple hashtag based pooling scheme working much better than the default way of unpooled tweets. When the dataset in consideration consists of generic terms simple hashtag pooling gives the best results in terms of topic coherence. On the other hand when we have tweets on specific named entities or events in general one might want to prefer hashtag assignment as it results in best PMI scores.

9 Related Work

Topic modelling is gaining increasing attention in different text mining communities. Latent Dirichlet Allocation (LDA) [1] is becoming a standard tool in topic modelling. 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. While the focus of this work was on blogs and search result snippets, it would be interesting to see how well they work on Twitter data. Also, the combination of the work proposed in this paper with the tweet pooling schemes we describe before could produce interesting results.

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. Our work is different from these in the sense that we provide a simple yet effective way which greatly improves the quality of topics obtained without making any major complicated modifications to standard LDA. The detailed experiments on a variety of datasets highlight which scheme works well for which kind of dataset and for which task.

10 Conclusion

In this paper we have proposed different tweet pooling schemes which produce topics which are representative of the events mentioned in the data and are coherent. With this work we aim to supportdeveloping topic models for short text and microblogs.

The Hashtag-based pooling scheme and tag assignment proposed works better than all other schemes across almost all datasets and metrics. It is encouraging that our simple algorithms produce good results across a broad spectrum of domains.

References