Improving LDA Topic Models for Microblogs via Automatic Tweet Labeling and Pooling

Author 1
Affiliation 1
Address 1
Country 1
email1

ABSTRACT

Twitter : the world of 140 characters poses serious challenges to the efficacy of topic models on short, messy text. While topic models such as Latent Dirichlet Allocation (LDA) have a long history of successful application to news articles and academic abstracts, they are often less coherent when applied to microblog content like Twitter. In this paper, we investigate methods to improve topics learned from Twitter content *without* modifying the basic machinery of LDA; we achieve this through various pooling schemes that aggregate tweets in a data preprocessing step for LDA. We empirically establish that a novel method of combining automatic hashtag labeling techniques with tweet pooling by hashtags leads to a vast improvement in a variety of measures of topic coherence across three diverse Twitter datasets in comparison to an unmodified LDA baseline and a variety of pooling schemes.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Theory

Keywords

ACM proceedings, LaTeX, text tagging

Introduction

In the general area of information retrieval or information access, the *undirected informational task* [**?**] 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 [**?**] and topic modeling have been developed to address. Probabilistic topic models such as Latent Dirichlet Allocation (LDA) [**?**] are a class of Bayesian latent variable models developed for analysing the semantic content of document corpora. Topic models uncover the salient patterns of a collection under the mixed-membership assumption: each document can exhibit multiple patterns to different extents. When analysing text, these patterns are represented as distributions over words, called *topics*. Topic models have been adapted to model document genres as diverse as news articles [**?**], blogs, academic abstracts [**?**], and encyclopaedia entries.

To address the undirected informational task arising for the exploration of Twitter content, we propose the use of popular topic models like LDA. However, Twitter content poses unique challenges different to much of standard NLP content:

- posts are short (140 characters or less),
- mixed with contextual clues such as URLs, tags, and Twitter names, and
- use informal language with misspelling, acronyms and non-standard abbreviations.

Hence, effectively modeling content on Twitter requires techniques that can readily adapt to this unwieldy data while requiring little supervision.

Unfortunately, it has been found that topic modeling techniques like LDA do *not* work well with the messy form of Twitter content [**?**]. 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. The naive application of LDA to Twitter content produces mostly incoherent topics — some are vaguely interpretable but contain unrelated words in the top-10 word set. For example, Table 1 demonstrates poor topic words as compared to topic words which are much more coherent and interpretable.

Poor Topics	Coherent Topics
barack cool apple health iphone	flu swine news pandemic health
los barackobama video uma gop	death flight h1n1 vaccine confirmed

Table 1: Sample Topic Words

How can we extract better topics in microblogging environments with standard LDA without the need of any major modifications? While lingistic "cleaning" of text could help somewhat, for instance [**?**], a complementary approach using LDA is also needed because there are so few words in a tweet. An intuitive solution to this problem is tweet pooling [**?**, **?**]: merging related tweets together and presenting them as a single document to the LDA model. In this paper we examine various tweet-pooling schemes and further enhancements to improve pooling quality. We compare the performance of these methods across different datasets; these are constructed

so that they are representative of the diverse collections of content possible in the microblog environment.

Evaluation of the resultant topic model on this data is challenging because of the unsupervised nature of the problem. Hence, we look at a variety of topic coherence evaluation metrics including the ability of the topics obtained to reconstruct known clusters and the interpretability of topics via statistical information measures.

Given our diverse datasets and evaluation metrics, we wish to answer the following questions:

- Do the different proposed pooling strategies perform better than unpooled tweets?
- Which pooling scheme works best for which metric and why?
- What further improvements can be made to the various pooling schemes so as to obtain topics which are coherent and interpretable?

In answering these questions, we make the following novel contributions:

- We show that different pooling schemes perform differently across different datasets and a few pooling schemes consistently outperform unpooled tweets.
- We show that a novel *hashtag-based pooling* approach leads to best results across *all* evaluation metrics. The performance is a substantial improvement over unpooled use.
- Given the prevalence of tweets without any hashtag, we present similarity metrics for automatic hashtag assignment for these tweets and demonstrate further improvement of hashtag-based pooling. We also observe that similarity metrics based on *inverse author frequency* (IAF) [**?**] offer some of the most robust performance for hashtag-based pooling across all datasets and evaluation metrics.

In the following sections we first provide a brief overview of topic modeling and LDA followed by a proposal for different tweet pooling schemes in Section 1. We next discuss Twitter dataset construction in Section 2 followed by evaluation metrics in Section 3. After discussing initial results in Section 4 where hashtag-based pooling emerges as a leading approach, we discuss further analysis and improvements to hashtag-based pooling in Section 5, followed by an in-depth evaluation of similarity metrics for automatic hashtag labeling. We present a final overall comparison of the best overall methods in Section 6. Related work is described in Section 7 and Section 8 summarises and concludes.

1. TWEET POOLING FOR LDA TOPIC MODELING

1.1 Latent Dirichlet Allocation and Topic Modeling

Latent Dirichlet allocation (LDA) [**?**] is a generative model for text. In LDA, each document may be viewed as a mixture of various topics. The generative process has documents represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. Learning the distributions (the set of topics, their associated word probabilities, the topic of each word, and the particular topic mixture of each document) is a problem of Bayesian inference, and many algorithms are available. We use the Mallet [**?**] implementation of LDA for performing our experiments.

1.2 Tweet Pooling Schemes

The goal of this paper is to obtain better LDA topics from Twitter content without modifying the basic machinery of standard LDA. As noted in Section , 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, these issues pose significant research challenges for LDA, which has been observed to perform poorly on Twitter [**?**].

To address these challenges, 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 an individual document does not present very rich term co-occurence data within documents that is highly useful for effective topic modeling. 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 a better topic model. We next describe an unpooled baseline LDA model and four different tweet pooling schemes.

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.

Author-wise Pooling: .

Pooling tweets according to author is the standard away of aggregating Twitter data to improve LDA topic modeling and has been previously proposed in the literature [**?**, **?**] and shown to be superior to unpooled Tweets. To use this method, we train LDA on aggregated author profiles, each of which combines all tweets posted by the same author.

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. 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 (potential trending topic) 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 $Mean(R) = \Sigma_y M(R, y)/z$. 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:

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

Let us denote the terms having burst-scores greater than 5 as *burst-term*. Then our first novel aggregation method of burst-wise polling aggregates tweets for each burst-term into a single docu-

ment trains LDA on the set of documents for each burst-term. The tweets which do not contain any of the burst-term were ignored for the purpose of training the LDA model. 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.

Temporal Pooling: .

The fourth scheme and our second novel pooling proposal is known as Temporal Pooling, which utilises the temporal information of the tweets. It is noted that whenever a major event occurs, a large number of users often 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 document of tweets posted within the same hour.

Hashtag-based 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. The tweet pool for each hashtag 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 with multiple hashtags being repeated across documents.

Other Pooling:.

While a few other combinations of pooling schemes (eg.author-time, hashtag-time, *etc*) are possible, the initial results obtained were not as good as those presented for the currently outlined pooling schemes. This may be due to the lack of data in each finer-grained pool. Despite the initial negative results, these combinations of pooling schemes might be further explored in future work and may help unveil even finer-grained topics (i.e., short-term events centred on an author group or set of hashtags).

2. TWITTER DATASET CONSTRUCTION

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 ($\sim$360K) in any of the twitter specific datasets, manual labeling of topics is not feasible. To circumvent this problem of unsupervised evaluation we carefully construct our datasets keeping the following point in mind: The datasets should cover diverse collections of content, but also the known source of the content should help in evaluation of the different schemes.

We construct three datasets which we believe are representative of the diverse collections of content found on Twitter. We chose one or two term queries (often with similar pairs of queries to encourage a non-strongly diagonal confusion matrix) to search a tweet collection and each resulting set of tweets was labeled by the query that retrieved it. 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:

Generic Dataset: 359478 tweets from 11 Jan'09 to 30 Jan'09. A general dataset with tweets containing generic terms which represent a broader sense.

Specific Dataset: 214580 tweets from 11 Jan'09 to 30 Jan'09. A dataset composed of tweets which have specific terms which refer to named entity topics.

Event Dataset: 207128 tweets from 1 Jun'09 to 30 Jun'09. An 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 two months (Jan'09 & Jun'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.*). Table 2 give the terms and the percentage tweets in the datasets which contain that term.

Dataset	Term/%
Generic	music/17.9 business/15.8 movie/14.5 design/10.8 food/9.6 fun/9.1 health/6.9 family/6.4 sport/4.9 space/3.2
Specific	Obama/23.2 Sarkozy/0.4 baseball/3.5 cricket/1.8 Mcdonalds/1.5 Burgerking/0.5 Apple/16.3 Microsoft/6.8 United-states/40.7 France/4.9
Events	Flight-447/0.9 Jackson/13.9 Lakers/13.8 attack/13.8 scandal/4.1 swine-flu/13.8 recession/12.3 conference/14.1 T20/4.4 Iran-election/8.6

Table 2: Datasets

Note that the three 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.

3. EVALUATING TOPIC MODELS - METRICS USED

Evaluation of the different topic models based on the features of coherence: topical consistency of documents assigned to a topic with high probability, or human interpretability of the most probable words for a topic are both important issues, 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 general approaches to measuring topical coherence, as well as a pure probability approach [**?**].

Clustering-based metrics:.

We would like to measure the quality of topics found by the models, that the models can recall known existing topics in the data, and can consistently assign the right tweets to the right topics. Fortunately, each dataset we constructed is a class-labeled dataset containing ten 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 and how consistent they are, we use clustering-based measures of purity and normalized mutual information (NMI), both defined below.

Query Term	Apple	Baseball	Burger King	Cricket	France	McDonalds	Microsoft	Obama	Sarkozy	USA
Apple	1	0.0007	0.0	0.0006	0.0003	0.0074	0.0008	0.0	0.0068	0.0001
Baseball	0.0036	1	0.0011	0.0	0.0	0.0	0.0038	0.0	0.0084	0.0
Burger King	0.0071	0.0	1	0.0	0.0013	0.0175	0.0	0.0	0.0	0.0049
Cricket	0.0002	0.0022		0.054	0.026	0.034	0.027	0.034	0.020	0.028
France					0.073	0.026	0.023	0.026	0.027	0.023
McDonalds						0.080	0.027	0.033	0.019	0.028

Table 3: Specific

Query Term	attack	conference	flight 447	iran election	jackson	lakers	recession	scandal	swine flu	t20
attack	1	0.0	0.0	0.0010	0.0008	0.0023	0.0	0.0	0.0015	0.0002
conference	0.0002	1	0.0	0.0010	0.0007	0.0023	0.0030	0.0015	0.0002	0.0
flight 447	0.0	0.0	1	0.0	0.0	0.0001	0.0	0.0	0.0	0.0
iran election	0.0018	0.0	0.0	1	0.0002	0.0004	0.0	0.0004	0.0	0.0
jackson	0.0337	0.0008	0.0	0.0	1	0.0350	0.0	0.0002	0.0007	0.0
lakers	0.0010	0.0012	0.0	0.0	0.0137	1	0.0	0.0001	0.0028	0.0
recession	0.0007	0.0015	0.0	0.0	0.0	0.0006	1	0.0001	0.0021	0.0
scandal	0.0006	0.0007	0.0	0.0008	0.0	0.0001	0.0002	1	0.0002	0.0
swine flu	0.0008	0.0008	0.0	0.0	0.0005	0.0035	0.0017	0.0004	1	0.0
t20	0.0118	0.0025	0.0	0.0	0.0010	0.0012	0.0010	0.0010	0.0	1

Table 4: dataset3-cf1

Query Term	business	design	family	food	fun	health	movie	music	space	sport
business	1	0.0118	0.0044	0.0036	0.0116	0.0076	0.0036	0.0027	0.0037	0.0035
design	0.0261	1	0.0031	0.0064	0.0101	0.0029	0.0031	0.0065	0.0054	0.0028
family	0.0094	0.0015	1	0.0089	0.0460	0.0083	0.0074	0.0091	0.0021	0.0028
food	0.0048	0.0031	0.0021	1	0.0014	0.0063	0.0032	0.0043	0.0029	0.0017
fun	0.0047	0.0041	0.0215	0.0035	1	0.0044	0.012	0.0041	0.0043	0.0024
health	0.0148	0.0089	0.0080	0.0152	0.0091	1	0.0022	0.0034	0.0045	0.0023
movie	0.0021	0.000044	0.0054	0.0019	0.0332	0.0008	0.0016	0.0102	0.0031	0.0072
music	0.0018	0.0034	0.0051	0.0025	0.0154	0.00		0.113	0.039	0.028
space	0.0081	0.0112	0.0010	0.0014	0.0171	0.0008	0.0093	0.0096	1	0.0016
sport	0.0450	0.0231	0.0120	0.0166	0.0290	0.0036	0.0096	0.0068	0.0104	1

Table 5: dataset2

Query Term	Apple	Baseball	Burger King	Cricket	France	McDonalds	Microsoft	Obama	Sarkozy	USA
Apple	0.516	0.028	0.026	0.028	0.021	0.029	0.025	0.027	0.016	0.024
Baseball		0.073	0.029	0.033	0.024	0.034	0.026	0.031	0.017	0.027
Burger King			0.141	0.031	0.024	0.036	0.024	0.030	0.018	0.026
Cricket				0.054	0.026	0.034	0.027	0.034	0.020	0.028
France					0.073	0.026	0.023	0.026	0.027	0.023
McDonalds						0.080	0.027	0.033	0.019	0.028

Table 6: Specific

Query Term	attack	conference	flight 447	iran election	jackson	lakers	recession	scandal	swine flu	t20
attack	0.077	0.035	0.030	0.031	0.031	0.029	0.036	0.030	0.026	0.026
conference		0.081	0.034	0.036	0.038	0.037	0.048	0.036	0.031	0.033
flight 447			0.023	0.034	0.029	0.027	0.039	0.031	0.029	0.026
iran election				0.016	0.030	0.029	0.037	0.031	0.024	0.028
jackson					0.061	0.035	0.039	0.031	0.027	0.027
lakers						0.083	0.039	0.031	0.025	0.027
recession							0.083	0.039	0.034	0.034
scandal								0.047	0.026	0.027
swine flu									0.017	0.023
t20										0.037

Table 7: dataset3

Semantic coherence and interpretability:.

Learnt topics should be coherent and interpretable. 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. It is possible to automatically measure topic coherence with near-human accuracy [**?**] using a score based on pointwise mutual information (PMI). We use this to measure coherence of the topics from different tweet-pooling schemes.

3.1 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

Query Term	business	design	family	food	fun	health	movie	music	space	sport
business	0.118	0.036	0.040	0.037	0.037	0.036	0.037	0.037	0.037	0.030
design		0.117	0.036	0.034	0.033	0.032	0.035	0.035	0.034	0.028
family			0.116	0.045	0.049	0.038	0.048	0.043	0.043	0.028
food				0.116	0.043	0.036	0.045	0.041	0.039	0.027
fun					0.118	0.033	0.045	0.041	0.040	0.024
health						0.115	0.035	0.034	0.035	0.030
movie							0.126	0.043	0.041	0.027
music								0.113	0.039	0.028
space									0.118	0.026
sport										0.101

Table 8: dataset2

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.

3.2 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. But high agreement is easy to achieve when the number of clusters is large, thus one needs a divisor to discount for a large number of clusters. The resulting two-part score is:

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

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 [**?**] 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.

3.3 Pointwise Mutual Information

One of the goals of our work is to get topics that are more coherent. PMI is one measure of the statistical independence of observing two words in close proximity. We treat two words as co-occurring if both the words occur in the same tweet. We compute the PMI of a given pair of words (w_i, w_j) as:

$$PMI(w_i, w_j) = log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} \quad (1)$$

For a topic w, we measure topic coherence using PMI defined as follows:

$$PMI\ \ Score(w) = \frac{1}{100}\Sigma_{i=1}^{10}\Sigma_{j=1}^{10} PMI(w_i, w_j)$$

where PMI Score(w) represents the PMI score of the topic w. The average of the PMI score over all the 10 topics is used as the final measure of the PMI score. A key aspect of this score is that it uses external data. [**?**] 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.

3.4 Held Out Probability

Another way of evaluating topic models is to compare predictive performance by estimating the probability of a subset of held-out documents. We used the Left to Right evaluation algorithm as described in [**?**] to calculate these values, which is an unbiased method. Another approach is the so-called document completion method [**?**], however with so few words we felt holding out a subset of a (small) document was ill-advised.

4. INITIAL RESULTS FOR POOLING SCHEMES

In this section we discuss the results of the experimental evaluation of the tweet pooling schemes introduced in Section 1. The datasets used were described in Section 2 while the evaluation metrics used were described in Section 3.

4.1 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. 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 9 presents the required statistics.

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 seven 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

Pooling Scheme	#of docs			Avg # of words/doc			Max # of words/doc		
	Generic	Specific	Events	Generic	Specific	Events	Generic	Specific	Events
Authorwise	208300	118133	67387	17.6	20.4	15.4	4893	3586	2775
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
Hashtag	8535	7029	4099	70.4	187.2	78.4	61918	420249	57794

Table 9: Document Characteristics for different schemes

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.

4.2 Comparison of Pooling Schemes

For the three datasets (viz. Generic, Specific and Events) and pooling schemes, we next evaluate the Purity scores, NMI scores, PMI scores and the Held-Out probabilities in Table 10 on the topic model obtained by training LDA using each scheme.

Based on these results we conclude that hashtag-based pooling scheme *clearly* 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-based pooling in detail and devise methods which further improve the results and provide better topics. We did not evaluate the held-out probability in later experiments as it agreed generally with the other scores, and is not as appropriate as a quality metric for the undirected information task.

5. 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.

5.1 How Many 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 11 presents few statistics on the percentage of tweets which do not have any hashtag in the three 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 11: % 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 participate in the hashtag pooling scheme. Since a large proportion of the available data is ignored, we need to figure out ways of incorporating this data while training LDA models. We first attempt addressing this issue with a naive brute force approach.

5.2 Incorporating Other Tweets :: Brute Force Approach

One way of incorporating the large portion of the dataset that is not aggregated into a pooled document could be to include all of these unpooled tweets as individual documents alongside the hashtag-pooled documents. Table 12 shows the results on the different datasets when using this method and the relative percentage improvement.

The results in the above table suggest this brute force technique substantially harms the topic coherence (lower PMI scores) but improves cluster reconstruction (higher purity scores). Hence, we need to look for alternative 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 all results so far.

5.3 Automatic Hashtag Labeling

Since a large number of tweets do not have hashtags, we conjecture that assigning *appropriate* hashtags to some of those tweets should help in improving our overall evaluation metrics – ideally all of purity, NMI, and PMI and not just a subset of these metrics as we saw for the brute force method. The more tweets without hashtags that are assigned at least one accurate hashtag, the more data will be incorporated into the correct pooled documents and the less data that will be discarded prior to running LDA. Next we present an algorithm for performing this automated hashtag assignment with high confidence.

5.3.1 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 conveniently have a collection of tweets that contain this particular hashtag. We will make use of this collection to compare each unlabeled tweet with each hashtag's collection and compute the similarity scores. If the similarity score between an unlabeled and labeled tweet exceeds a certain threshold C, we assign the hashtag of the labeled tweet to the unlabeled tweet (and hence it joins the pool for this hashtag). The similarity metric we use initially is just the cosine similarity between the term frequency (TF) vector space model (i.e., $t_i := f_i$ if the frequency count for term i in tweet t is f_i) of two Tweets t and $\hat{t}$ where cosine similarity is simply $Sim(\hat{t}, t) = \frac{\hat{t} \cdot t}{\|\hat{t}\| \|t\|}$. After initial experiments with the cosine TF similarity, we will later define and experiment with additional similarity metrics. Algorithm 1 describes the hashtag assignment procedure; the document resulting from pooling tweets for each hashtag h under this assignment scheme will then be composed of $T_h \cup \hat{T}_h$.

5.3.2 Automatic Hashtag Labelling

Table 13 presents the results of hashtag assignment based on the three metrics for different confidence thresholds C 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

Scheme	Purity			NMI Score			PMI score			Held Out Probability		
	Generic	Specific	Events	Generic	Specific	Events	Generic	Specific	Events	Generic	Specific	Events
Unpooled	0.49	0.64	0.69	0.28	0.22	0.39	-1.27	0.47	0.47	-82.2	-89.3	-86.3
Author	0.54	0.62	0.60	0.24	0.17	0.41	0.21	0.79	0.51	-63.0	-68.6	-66.4
Hourly	0.45	0.61	0.61	0.07	0.09	0.32	-1.31	0.87	0.22	-64.8	-69.4	-67.9
Burstwise	0.42	0.60	0.64	0.18	0.16	0.33	0.48	0.74	0.58	-56.7	-59.0	-57.8
Hashtag	**0.54**	**0.68**	**0.71**	**0.28**	**0.23**	**0.42**	**0.78**	**1.43**	**1.07**	**-55.9**	**-58.9**	**-55.4**

Table 10: Initial results of different pooling schemes

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

is intuitive: on increasing the threshold, fewer 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.

The results obtained via hashtag assignment are better than those of simple hashtag-based pooling as well as the unpooled scheme. Encouraged by this, we next look at some other similarity metrics and see if we could further improve our results.

5.3.3 Other Similarity Metrics

There is no inherent reason why cosine TF should be the ideal similarity metric for automatic labeling of Tweets and hence in this section we define a number of other similarity metrics and evaluate them on hashtag pooling with automatic hashtag labeling. We now describe the following five document representations to be used to be used in conjunction with the cosine similarity function Sim defined previously:

Term Frequency (TF): As previously defined for TF, we set the vector element $t_i := f_i$ if the frequency of term i in tweet t is f_i.

Inverse Document Frequency (IDF): Given the occurrence of term i in N_i of the total N tweets, IDF_i, the ith entry of t's document representation (t_i) is set as $t_i := IDF_i = \log \frac{N}{N_i}$. Thus the IDF of a rare term in the corpus is high, whereas the IDF of a frequent term is low.

Inverse Author Frequency (IAF): The relationship of terms and authors could be harnessed using the inverse author frequency (IAF) [**?**], which is computed similarly to IDF except that we let A_i represent the number of authors who mention term i out of the total number of authors A; $t_i := IDF_i = \log \frac{A}{A_i}$.

The IAF of a term used only by a few different authors is high, whereas the IAF of a term used by most authors is low. In our framework we use the IDF and IAF values to weigh the boolean vector space model (0-1 in place of TF) and compute IDF- and IAF-weighted inner products on the boolean vector spaced model.

TF-IDF: The tweet document representation is simply the elementwise product of TF and IDF: $t_i := f_i \cdot \mathrm{IDF}_i$.

TF-IDF-IAF: The tweet document representation is simply the elementwise product of TF, IDF, *and* IAF: $t_i := f_i \cdot \mathrm{IDF}_i \cdot \mathrm{IAF}_i$.

We experimented with other similarity metrics, but found these five to offer the best performance, hence we show their evaluation next.

Figure 1 presents the results obtained for different similarity metrics for automatic hashtag labeling in the hashtag pooling scheme discussed above on the same three datasets and evaluation metrics used previously. It is interesting to note that the simplest of all, TF-based cosine similarity, performs well in almost all cases except one. We note that metrics like IAF that are relevant only on socially authored media such as Twitter perform quite well on this task. Indeed IAF often performs quite comparably to or better than TF and is quite consistent across all three datasets, all three evaluation metrics, and all confidence thresholds, unlike TF and other metrics which show cases of poor performance or anomalous dips. Hence, when compared to the other metrics, we might conclude that IAF is a robust similarity metric for purposes of automatic hashtag labeling as demonstrated consistently across all these experiments.

Algorithm 1: HASHTAG ASSIGNMENT

input : H: set of hashtags
T_h : for each $h \in H$, set of tweets labeled with h
T_n : set of tweets with no hashtag
$Sim(\cdot, \cdot)$: a similarity function between two tweets
C : Threshold on the similarity score
output: $\hat{T}_h$: for each $h \in H$, set of tweets newly assigned label h

```
begin
    foreach h ∈ H do
        T̂_h := ∅
        foreach t̂ ∈ T_n do
            // Hashtag h assignment for t̂ if any t ∈ T_h is near:
            foreach t ∈ T_h do
                if Sim(t̂, t) > C then T̂_h := T̂_h ∪ {t̂}, break
    return T̂_h
end
```

6. OVERALL COMPARISON

The goal of this work was to get better topics which have better coherence without modifying the basic machinery of standard

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
1.0 (Simple Hashtag Pooling)	0.54	0.68	0.71	0.28	0.23	0.42	0.78	**1.43**	**1.07**

Table 13: Hashtag Assignment Results

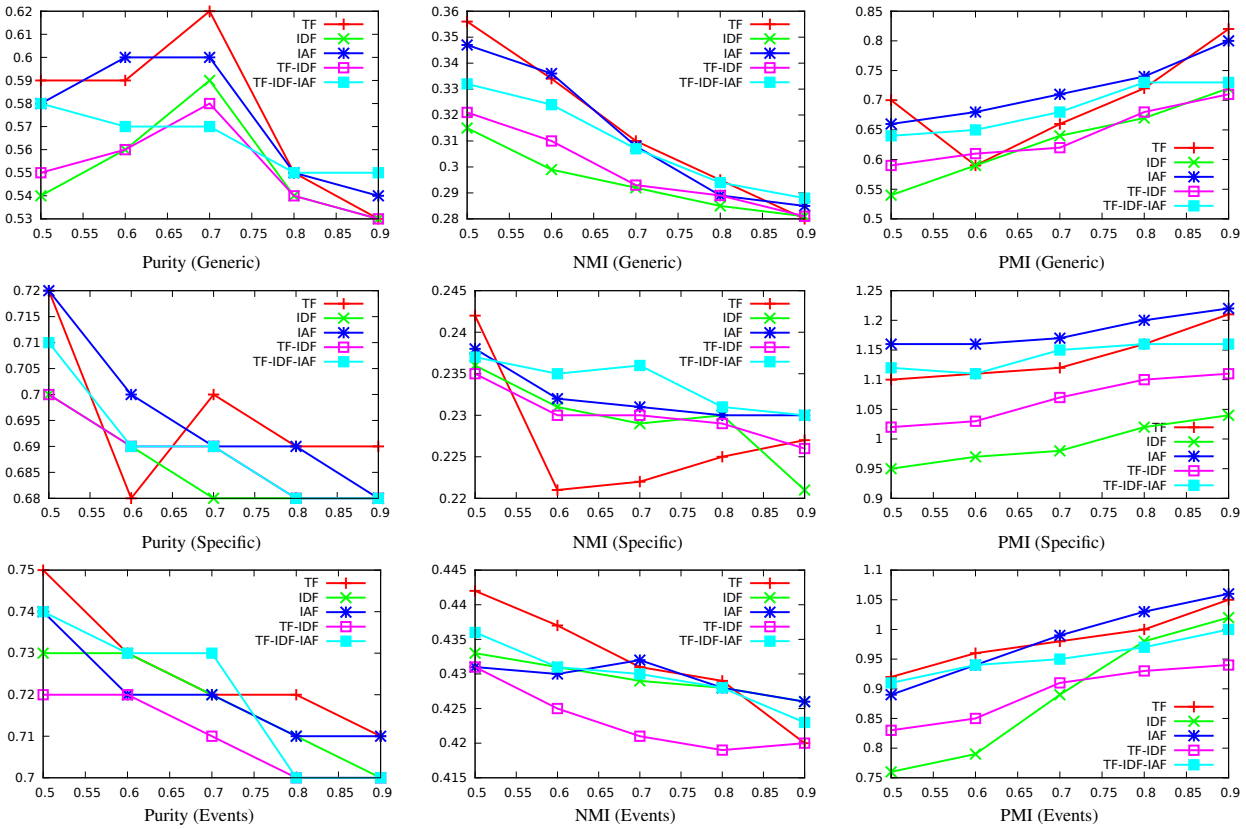

Figure 1: Comparison of Purity, NMI and PMI scores (y-axis) for five different similarity metrics (TF, IDF, IAF, TF-IDF, TF-IDF-IAF) for various Hashtag assignment confidence threshold limits (x-axis).

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 best of the different schemes: unpooled vs simple hashtag vs hashtag assignment on the three datasets and across the three metrics.

In Table 14, we see that the best Purity and NMI scores are obtained by hashtag assignment while even the simple hashtag-based pooling works much better than the baseline method 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 for the Specific and Events datasets, presumably because it is better at recovering coherent multiword topics for these entities/events.

7. RELATED WORK

Topic modeling is widely used in text mining communities with LDA being the benchmark. LDA has been extended in a variety of ways, and in particular for social networks and social media, a number of extensions to LDA have been proposed. For example, [**?**] 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 [**?**] with the tweet pooling schemes we describe before could produce interesting results. For automatic hashtag labeling that proved crucial to improving topics in our hastag-based pooling model, [**?**] also uses tweet similarity as a criteria, but does not ex-

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

plore metrics based on inverse author frequency [**?**] that we found to offer the most robust performance across datasets and evaluation metrics. Additional features for hashtag assignment can be found in the comprehensive study [**?**] which can be leveraged in future extensions.

Our work is quite different from many pioneering studies on Twitter and topic modeling because we focus on how we could get better topic coherence over tweets with minimal modification to existing models. Prior work on topic modeling for tweets includes the work of [**?**] which presents a scalable implementation of a partially supervised learning model (Labeled LDA). [**?**] empirically compare the content of Twitter with a traditional news medium, New York Times, using unsupervised topic modeling. [**?**] use the topic modeling approach for predicting popular Twitter messages and classifying Twitter users and corresponding messages into topical categories. The TwitterRank system [**?**] and [**?**] uses author-based pooling to apply LDA to tweets. [**?**] compared topic characteristics between twitter and traditional news media; they propose to use one topic per tweet (similar to PLSA), and argues that this is better than no pooling, or the author-topic model. [**?**] used LDA for tweet retrieval. In addition, they used retweet as an indicator of "interestingness" to improve retrieval quality, which suggests additional features we could incorporate in future extensions to our pooling framework.

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 our novel contribution of hashtag-based pooling and automatic hashtag labeling using similarity metrics like IAF [**?**] as an approach that improves a range of topic coherence measures.

8. SUMMARY AND CONCLUSION

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 10 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 13) 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 13 highlight that topic coherence increases down the column with a threshold of 0.9 giving most coherent topics. Overall, results shown in Table 14 compare unpooled scheme with simple hashtag pooling and hashtag assignment schemes. Across diverse datasets and various topic coherence metrics, the best hashtag assignment method performs substantially better in comparison to an unmodified LDA baseline and a variety of existing and novel pooling schemes. This indicates the promise of this novel automatic hashtag labeling and pooling approach that drastically improves the quality of topic models for Twitter and microblog data.

9. REFERENCES