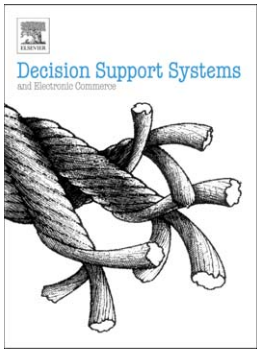


Tracking Geographical Locations using a Geo-Aware Topic Model for Analyzing Social Media Data

Marianela García Lozano, Jonah Schreiber, Joel Brynielsson

PII: S0167-9236(17)30084-2
DOI: doi:10.1016/j.dss.2017.05.006
Reference: DECSUP 12841

To appear in: *Decision Support Systems*

Received date: 20 September 2016
Revised date: 13 April 2017
Accepted date: 4 May 2017

Please cite this article as: Marianela García Lozano, Jonah Schreiber, Joel Brynielsson, Tracking Geographical Locations using a Geo-Aware Topic Model for Analyzing Social Media Data, *Decision Support Systems* (2017), doi:10.1016/j.dss.2017.05.006

Tracking Geographical Locations using a Geo-Aware Topic Model for Analyzing Social Media Data

Marianela García Lozano[a,b,*], Jonah Schreiber[b,c], Joel Brynielsson[a,b]

[a] *FOI Swedish Defence Research Agency, Stockholm, Sweden*
[b] *KTH Royal Institute of Technology, Stockholm, Sweden*
[c] *Google, Inc., Mountain View, California*

Abstract

Tracking how discussion topics evolve in social media and where these topics are discussed geographically over time has the potential to provide useful information for many different purposes. In crisis management, knowing a specific topic's current geographical location could provide vital information to where, or even which, resources should be allocated.

This paper describes an attempt to track online discussions geographically over time. A distributed geo-aware streaming latent Dirichlet allocation model was developed for the purpose of recognizing topics' locations in unstructured text. To evaluate the model it has been implemented and used for automatic discovery and geographical tracking of election topics during parts of the 2016 American presidential primary elections. It was shown that the locations correlated with the actual election locations, and that the model provides a better geolocation classification compared to using a keyword-based approach.

Keywords: social media, topic modeling, geo-awareness, trend analysis, latent Dirichlet allocation, streaming media

[*] Corresponding author

Email addresses: `mgl@kth.se` (Marianela García Lozano), `jonahsc@kth.se` (Jonah Schreiber), `joel@kth.se` (Joel Brynielsson)

1. Introduction

The rapid growth of social media has enabled millions of people to express themselves and reach wide audiences. As the quantity of openly available text on the internet is increasing, so is the interest in categorizing such data and to capture overviews of ongoing discussions. Such summaries of public opinion have found broad uses in industry, commerce, and security. For example, disaster relief supplies can be better distributed using knowledge compiled by systems that analyze Twitter trends for areas of concern [1].

Topic modeling is an emerging machine learning field that aims to arrange large corpora of text into representative topics [2]. These topics describe thematic patterns and relations between texts. Most topic models offer unsupervised learning, i.e., they discover topical structures by statistical methods and do not require manual per-document annotations of the "correct" classification. They are typically trained on a training corpus before being applied to new and unexplored documents, but the training itself can also reveal interesting morphologies of the corpus.

The use of topic models on social media is still in its infancy, and many of the applications are yet unknown. Topic models can track trends in a real-time, online fashion where the term "trend" is defined to mean distinct temporal changes that a topic experiences. It does not necessarily indicate popularity, but rather topic evolution. Trend detection and tracking is a very valuable capacity to decision-makers, because it can suggest how to best allocate limited resources.

A key feature of online trends is that many have distinctly evolving geographical locations. For example, touring musicians visit different cities over the duration of their tour, which tends to be temporally reflected in relevant online discussions. However, automatic extraction of locations from such discussions is difficult. Some media let users report their location, called geo-tagging, but such data is very often unavailable or unreliable. To solve this, some previous work has focused on making topic models geo-aware. The term "geo-aware" is defined here to mean that a model recognizes that text content may include explicit or implicit geographical locations. Previous work explores mostly if the location of a document's author can be identified. Less work has been done to identify how geographical locations correlate with topics, and even less on how topical locations change over time.

This paper addresses this latter problem. We seek not only a compounded model of topics and locations, but a more integrated solution. It should not only correlate topics with locations, but also recognize that topics and locations might not be independently associable. Some topics, like international relations, are inherently geographical while others, like mathematics, are not. We need a model that is sensitive to such knowledge and incorporates it in a sophisticated way as it detects emerging geographical trends.

1.1. Purpose

The study presented herein has served to design and evaluate a model for tracking online discussions with regard to geographic correlation over time. Textual-based geographical locations are ambiguous and therefore inherently contextual, and context is also difficult to model since texts can have complex semantics. The challenge is therefore not only to recognize places, but also to disambiguate them whilst in the presence of words related to a complex configuration of topics.

2. Background

This section contains a brief overview of related work and provides a historical perspective concerning the primary techniques that this paper builds upon.

2.1. Geographical Awareness

During the last couple of years, location association in social media has been a topic of much interest. When discussing location in this context it has mainly been focused on finding out the account holder's location. The geo-positioning methods have employed a number of indirect indicators such as looking at the social network, the style of writing (choice of word, etc.), or explicit location references given in the text content [3, 4, 5, 6]. Combining these types of location indicators with metadata such as timezone, language, and even posting time, has resulted in fairly good localization of both user residence and tweet post location [7, 8]. However, the results of these efforts usually do not differentiate between the account holder's current or past locations, and they also do not focus on the topical correlation.

Several studies claim that only about 0.5–2% of tweets are explicitly geo-tagged by the users [8, 9, 10]. Hence, some efforts in inferring locations have focused on geo-positioning untagged posts [10, 11].

A few have explored the correlation between geographical location and topics [4, 12]. However, these results mainly focus on identifying the topics that are present at a specified geographical area, and the implementations solely consider locations extracted from Twitter post location fields (i.e., where the tweet is from) and user self-proclaimed home locations rather than locations extracted from the tweet content. Hence, these approaches will only make use of a few thousands of the tweets that are explicitly geo-tagged by the users, and one must also take into account that i) the explicit geo-tagging of a tweet can easily be faked, ii) users' self-proclaimed home location is of highly varying quality since everything from "here" to "Mordor" is allowed.

Related to this is the study by Wang et al. who modified a topic model to mine geographical locations [13]. They used blogs and news data sets to study the correlation between topics in the form of words and geolocations. However, they did not look at these correlations over time. Another study has actually focused on topic tracking and correlating it with geographical prevalence [14]. In this case, however, a specific topic to be followed (the flu) is chosen and mainly relies on geo-tagged tweets and the users' self-proclaimed residence. Yet another interesting and related effort is presented by Tsou et al. [15]. Besides Twitter data these authors also make use of search results emanating from the web search engines Yahoo! and Bing as input in order to monitor temporal topics with spatial prevalence. The work contains a case study where the two main candidates during the 2012 American presidential elections (i.e., Barack Obama and Mitt Romney) are tracked. However, the effort is supervised and keyword-based in that the authors manually chose the topic, i.e., the presidential candidates, and the locations, i.e., the 50 largest cities in the U.S.

As we have seen there have been several efforts to automatically deanonymize the geolocation of the user, i.e., the account holder's residence. Further efforts have also gone into finding out the tweet's geolocation, i.e., where the tweet was sent from. In this paper we focus on a third type of geolocalization that so far, to the best of our knowledge, has been largely unexplored. We focus on the geolocation of the tweet's topic, i.e., the geolocation of what the tweet is about, and follow it over time.

2.1.1. Geographic Location Extraction in Text

Throughout this paper we focus on very short texts (tweets are only 140 characters long) and aim to find the location(s) associated with the topic of the text. As the topics evolve over time we track their changing locations. Such locations are usually directly observable (non-latent); while texts may use location synonyms that require disambiguation (for example, *Central Park*), locations are rarely inferred rather than observed. The problem is hence reduced to *recognize* which words feature a location in a text, a problem in the domain of *Named Entity Recognition* (NER). NER is a subdomain of information extraction with the purpose of identifying and classifying which words in a text represent predefined "entity" classes, i.e., persons, locations, organizations, quantities, etc.

The simplest method of extracting locations from a text would be to use a gazetteer, which in its simplest form is a list of words consisting of geographical locations. The list is then used to find occurrences of these names in text. The disadvantages are that the list implies a closed world assumption, i.e., new places of interest would not be recognized, and that it is not context sensitive, meaning that it does not differentiate between the location "Paris" and the person "Paris" or "a white house" and "the White House." Further, building such a list requires much labor and the list might also contain errors. These kinds of list are a good starting point, though, when training a model to automatically recognize locations in a text. For examples of gazetteer based NER methods, see [16, 17, 18].

Another method to identify locations in text is to use pattern matching and learn rules like "I went to X" or "the town X," by utilizing tokens near the entity as location indicators. However, such rules tend to result in false positives, and it is also difficult to come up with an exhaustive set of rules. For examples of rule-based NER methods, see [19, 20, 21].

As previously mentioned, disambiguation of geographical names is a problem that is closely related to the challenge of performing geolocation using social media data. An interesting method for managing disambiguation using preference learning is the method presented by Zhang et al. for performing geographical NER in Twitter messages [22]. These authors achieved better precision than state of the art competitors, but did not study any topical correlation.

Stochastic approaches perform better across domains. Many NER methods aim to provide *structured prediction*, where probabilistic models are used to predict words' entity class membership. The *conditional random field* (CRF) model, a graphical discriminant model, has been used in many state of the art classification solutions [23]. In our model we use CRFs to classify entities of interest to our task, see Section 3.2 for more details.

2.2. Topic Modeling

Topic modeling is a relatively young machine learning field (∼15 years) and one of the earliest algorithms is probabilistic latent semantic indexing (PLSI) [24]. Originally, the main motivation for researching topic models was in the interest of automatically annotating documents with summarizing information. With time, however, interests in other applications, such as sentiment analysis, detection of emerging trends, bioinformatics, opinion classification, and big data analysis, have emerged [25, 26].

In 2003 Blei et al. presented their paper on latent Dirichlet allocation (LDA) which solved many of the earlier topic models' issues [27]. Here, each document is assumed to be produced from a proper generative probabilistic model, which uses Dirichlet distributions to sample topic and word mixtures. Documents feature a mixture of latent topics, which in turn feature a mixture of words. Hence, LDA allows a flexible specification of prior beliefs about a corpus, and inferring the latent topical structure is natural. LDA is readily extensible and much related work in topic modeling has been focused on modifying LDA to recognize either more general or more specific text features, e.g., hierarchical Dirichlet processes (HDP) which does not require a prior topic count belief but would instead infer the number of topics in a document [28], an online (streaming) version that can handle growing corpora [29], and a model which bisections topical knowledge into general and specific aspects [30].

In this paper we focus on trends, i.e., topic models that evolve through time. One such algorithm is the time discretized online LDA extension (OLDA) [31]. It uses results from earlier models as prior knowledge for current models. The work showed that the model could be built incrementally and was efficiently scaleable, and it also detects emerging topics in real-time. Another model is the online variational Bayes method that has been shown to be faster than online algorithms using sampling [32]. A development of the OLDA model to maintain a dynamic vocabulary introduced a damping factor to the propagation of learned priors [33].

Deep learning is currently the state of the art within several fields. Hence, researchers have begun exploring topic modeling using neural networks, vector models (word2vec), and also combining these techniques with LDA [34, 35, 36]. However, these efforts were deemed too complex and differently focused from what this project aimed to do.

2.2.1. Topic Models and LDA

Topic models assume that the content of documents is governed by abstract *topics* which are hidden (or *latent*). Usually, topic models assume that each document is a bag of words (lexical units) and that the order of words is unimportant. Most models also assume that each document is assigned a number of topics and that documents are otherwise independent. LDA assumes that a fictive and stochastic generative process is producing each document [27]. Each document may feature a *mixture* of topics. LDA can thus learn not only how words co-occur but also how topics co-occur. Care must however be taken to parameterize LDA judiciously.

LDA uses four central parameters: K, the number of topics; V, the size of the vocabulary; and the vector *sparsity parameters* $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_K)$ and $\beta = (\beta_1, \beta_2, \ldots, \beta_V)$. α and β (sometimes called *Dirichlet priors*) parameterize the Dirichlet distributions $\mathrm{Dir}_K(\alpha)$ and $\mathrm{Dir}_V(\beta)$ from which in turn θ and φ can be sampled. θ represents the proportions of featured topics in a document d and, similarly, φ represents the proportions of words w that a topic tends to feature. $z_{d,w}$ is the topic of word w in document d.

Ultimately, the problem is to infer the posterior distributions φ and θ, given the observed words as well as $\mathrm{Dir}_K(\alpha)$ and $\mathrm{Dir}_V(\beta)$. Only the words are observed, so even though α and β are latent, they must be presumed (initialized) to perform this inference. Approximate inference of the posterior is typically used. In our work we have used Gibbs sampling, see [37].

LDA's readily extensible nature lends naturally to modeling temporal events and trends. AlSumait et al. introduced mathematics for creating an *online* LDA (OLDA) model in 2008 which uses time discretization and sliding windows to update the model as documents arrive in temporal batches [31]. Lau et al. progressed this idea to use a dynamic vocabulary and to update the priors α and β of new time slices using θ and φ from the previous time slice [33]. This model, which we denote *streaming* LDA (SLDA), provides useful methods to detect emerging trends and is the one we use in this paper[1]. This extension to LDA has several advantages. SLDA's time complexity is independent of the number of time slices, whereas LDA grows linearly for the whole corpus and thus quadratically overall. SLDA is sensitive to emerging trends whereas LDA converges due to a growing corpus. Finally, SLDA has a dynamic vocabulary allowing detection of trendy words and a more accessible interaction with NER models.

3. Methodology

This section covers the methodology utilized, primarily code, libraries, interfaces, and frameworks. The main workflow used Twitter as a data set source, but Twitter could handily be replaced by other streamable (or time-ordered offline) sources. This workflow was further divided into four distinct phases, as illustrated in Figure 1. Each phase is detailed in a section below.

3.1. Tweet Collection

Twitter provides access to its documents (tweets) by means of the *Twitter API* [39]. Apache Flume, which uses this API, was used to collect certain sets of tweets. It could readily stream large amounts of tweets and store these onto an Apache Hadoop-compliant distributed resilient storage. Flume was run on a cluster of five worker nodes, each with 2 × Intel E5-2620v2 2.1 GHz 6-core CPUs and 32 GB of RAM. Twitter4J was also used in a Java implementation to directly access the Twitter API and download tweets from specific users. The following sets covering parts of the American presidential primary elections were compiled:

- **USTwiNews**. The last 3,200 tweets from large news networks' Twitter handles were collected on April 27th, 2016 using Twitter4J. Wikipedia's

[1] SLDA as an abbreviation has previously also been used in the field of computer vision, but there it stands for *Spatial* LDA [38].

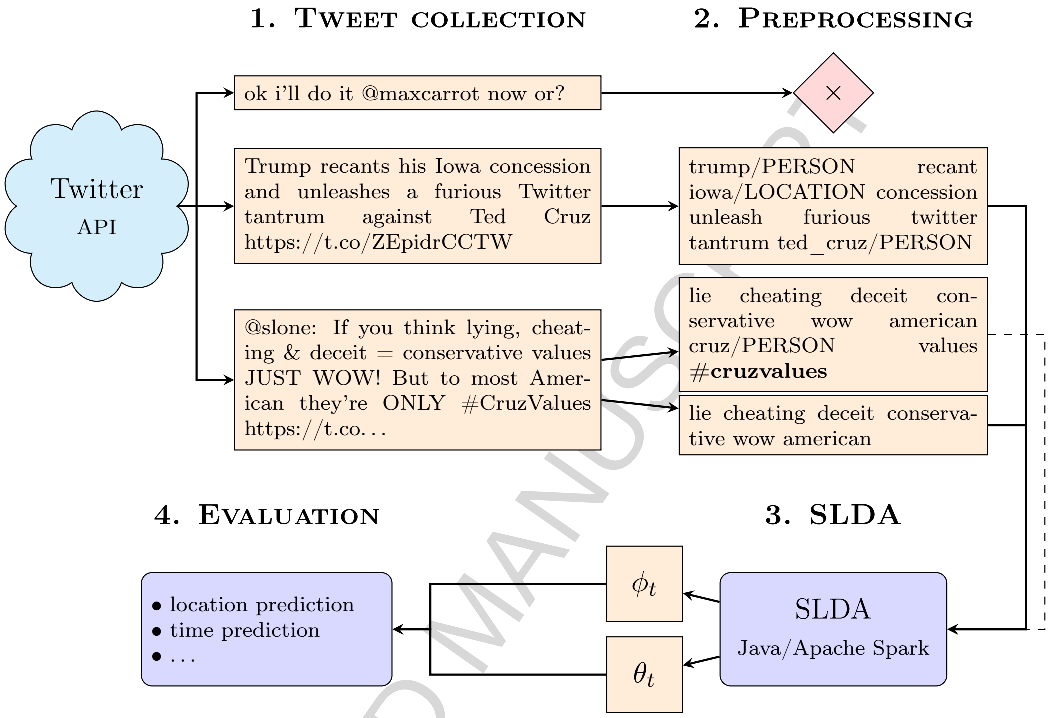

1. Tweet collection
2. Preprocessing
Twitter
API
ok i'll do it @maxcarrot now or?
×
Trump recants his Iowa concession and unleashes a furious Twitter tantrum against Ted Cruz https://t.co/ZEpidrCCTW
trump/PERSON recant iowa/LOCATION concession unleash furious twitter tantrum ted_cruz/PERSON
@slone: If you think lying, cheating & deceit = conservative values JUST WOW! But to most American they're ONLY #CruzValues https://t.co...
lie cheating deceit conservative wow american cruz/PERSON values #cruzvalues
lie cheating deceit conservative wow american
4. Evaluation
3. SLDA
• location prediction
• time prediction
• ...
ϕ_t
θ_t
SLDA
Java/Apache Spark

Figure 1: Flowchart of the methodology workflow. Different data sets and different parameters were used as input to SLDA, as indicated by the dashed line exemplifying how tweets either had their hashtags removed or split and copied to the end (as explained further in Section 3.2).

pages on *Television News in the United States*, *List of Newspapers in the United States*, *News magazine* and *Streamy Awards* were consulted to find the most popular news companies for television, newspapers, news magazines and internet news, specifically. This resulted in a list of 31 handles such as `@abc`, `@abcnews`, `@cbsnews`, `@nytimes`, `@TheYoungTurks`, and so on.

- **USE2016**. Between February 1st and May 1st 2016, approximately 144 million tweets were collected using common United States 2016 primary election hashtags and keywords, such as `#election2016`, `#2016election`, `#trump`, `#bernie`, `#hillary`, `#bush`, etc.

Table 1 summarizes some statistics for this set.

	USTwiNews	**USE2016**
Theme	US/World News	US election keywords
Size	277 MB	650 GB
No. of documents	138,000	144,000,000
Retweet/tweet ratio	N/A	$\sim 3/1$
Time period	27 Jan – 27 Apr 2016	03 Feb – 02 May 2016

Table 1: Statistics for compiled data sets.

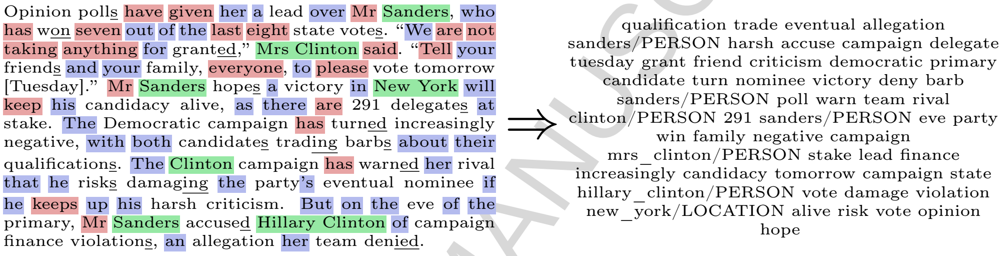
3.2. Preprocessing

For experimental purposes a number of different subsets were created. For example, data sets tended to have very large amounts of retweets, and so subselections were made where retweets were either eliminated completely, or heavily reduced. These subselections are detailed below:

- Retweets were either eliminated completely, or reduced in count to $\sqrt{N}$ of the number of retweets N of the original tweet.
- Tweets either had their hashtags removed or kept. If kept, hashtags were copied to the end of the tweet and split in place. For example, `Monday's #TedCruz campaign` would become `Monday's Ted Cruz campaign #TedCruz`.
- To keep the document count tractable, sampled selections of tweets could be taken from tweet sets; for example, randomly (uniformly) choosing 10% of the documents.

The sets of tweets were accessed using a Java program developed to use Apache Spark [40]. Spark is a distributed/cluster computing solution with very useful features for resilience, fault-tolerance, integration with Hadoop, and inherent parallelism. This program allowed processing to take place in parallel, with documents split amongst worker nodes (processors), and hence work much faster.

After dividing the tweets into several experiment sets using Spark, the preprocessing was mainly performed (still with Spark in parallel) using Stanford NLP's package for natural language processing [41]. This package includes very useful tools such as tokenization, sentence splitting, lemmatization, stemming, classification, annotation, and more. Figure 2 shows an example. First, tweet metadata was used to remove uninteresting entities from the text, specifically URL links, user mentions, multimedia, and possibly hashtags. Then, the remaining text was stripped of uninteresting symbols (such as currency signs, Unicode emoticons, etc.) while keeping sentence delimitation symbols (comma, period, etc.)



Opinion polls have given her a lead over Mr Sanders, who has won seven out of the last eight state votes. “We are not taking anything for granted,” Mrs Clinton said. “Tell your friends and your family, everyone, to please vote tomorrow [Tuesday].” Mr Sanders hopes a victory in New York will keep his candidacy alive, as there are 291 delegates at stake. The Democratic campaign has turned increasingly negative, with both candidates trading barbs about their qualifications. The Clinton campaign has warned her rival that he risks damaging the party’s eventual nominee if he keeps up his harsh criticism. But on the eve of the primary, Mr Sanders accused Hillary Clinton of campaign finance violations, an allegation her team denied.

⇒

qualification trade eventual allegation
sanders/PERSON harsh accuse campaign delegate
tuesday grant friend criticism democratic primary
candidate turn nominee victory deny barb
sanders/PERSON poll warn team rival
clinton/PERSON 291 sanders/PERSON eve party
win family negative campaign
mrs_clinton/PERSON stake lead finance
increasingly candidacy tomorrow campaign state
hillary_clinton/PERSON vote damage violation
new_york/LOCATION alive risk vote opinion
hope

Figure 2: An excerpt from a BBC news article showing preprocessing into a bag-of-words representation. Words highlighted in blue are removed as they belong to an uninteresting word class, such as prepositions. Words in red are stop words and are also removed. Words in green are a recognized named entity. Underlined letters are modified due to lemmatization reducing words to base form. To the right of the arrow is the resulting preprocessed bag-of-words. Final word order does not matter.

This text was then put through Stanford NLP’s conditional random field classifier (CRFClassifier), trained on the CoNLL 2003 training data with 3 classes: `LOCATION`, `PERSON`, and `ORGANIZATION`. The phrases annotated by these classes were stored separately, and fed through an NLP pipeline performing tokenization, sentence splitting, and lemmatization. This produces a list of lowercase words where each word has been reduced to its base form (for example, *lying* becomes *lie*) and where each word has been annotated with a tag describing the type of word (for example, noun, verb, or cardinal number).

Only certain words were kept (for example, prepositions were discarded), and, in addition, a list of 682 very common words (“stop words”) was used to filter out less interesting words [42]. Finally, this list was amended with the classification done earlier, so that classified phrases (such as *san francisco*) become “words” (tokens) themselves (*san_francisco/LOCATION*). These preprocessed bag-of-words were then stored on distributed storage in a simple text format, with only the tweet ID and the tweet's timestamp as metadata.

3.3. SLDA

An SLDA implementation, as described in Section 2.2.1, was developed in Java, again utilizing Apache Spark. First, a streaming source of tweets is specified, which may either be an offline set of possibly preprocessed tweets stored with Hadoop, or a live stream of tweets (in which case preprocessing happens live). SLDA parameters (α, β, K, window width, etc.) are also specified here. Table 2 shows typical parameters used in SLDA.

K	α	β	Gibbs iter.	$V_{\text{keep top}}$	Time slice	Window width	c
30	0.02	0.01	1,000	15,000	24 hours	4	0.5

Table 2: Typical SLDA parameters for data sets. They are constant over a stream.

Then, in a streaming fashion, a vocabulary is created for each window of preprocessed tweets. Spark frames each window as a *resilient distributed data set* (RDD), on which distributed data operations are possible. In effect, each worker node manages a partition of the current window of documents. Words are counted in parallel using the RDD operations `flatMapToPair: words -> {(word, count)}`, `reduceByKey: (count1, count2) -> count1 + count2`, and `top(V)`, selecting the V most common words. The vocabulary then provides means of converting between an actual word (a string) and an index $0 \leq v < V$.

At this point, the SLDA model has its priors α and β initialized. Special initialization thus happens if a previous model exists (for all but the first window). This is done partly in parallel using Spark's `leftOuterJoin` to copy over previous θ and z for initialization of α. Previously seen documents keep their old z assignment. β is initialized serially. β is then copied to all worker nodes by means of Spark's `Broadcast`.

Then, the Gibbs iterations start, resampling z in parallel for each document by using θ and φ. However, here a very important assumption needs to be made.

As depicted in Figure 3, Gibbs sampling is an inherently serial process because changing z changes φ, which in turn influences the posterior distribution over the assignments of words to topics $P(z|w)$. Hence, φ must then be synchronized every time z changes. Reading φ and writing z must be mutually excluded, leading to virtually no parallel speedup.[2] Newman et al. reason, however, that this interdependence is weak, because the number of worker nodes P is often much smaller than the number of documents D ($D \gg P$) [43]. In effect, instead of sampling asymptotically from the true posterior, one samples from an approximation of the posterior. Hence, they assume that synchronizing less often on φ creates an insignificant error and that a useful model can still be learned.

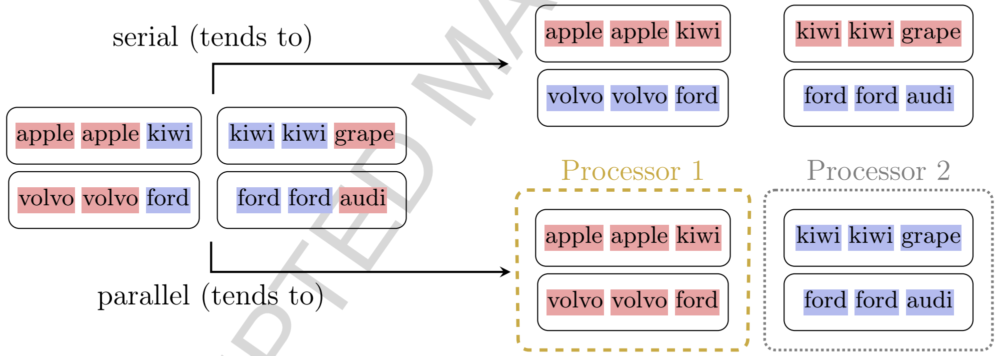


Figure 3: z interdependence problems when $D \sim P$ with low α, 4 documents, 2 topics (red and blue) and 6 words. The topics start out fully confounded. Serial sampling tends to separate the topics while parallel sampling tends to confound the topics. This is because the resampled topic of *kiwi* in Processor 1's document is unavailable to Processor 2. As such, it doesn't know to bias the coloring of *kiwi* to that of Processor 1's. In this example, serial Gibbs in one iteration correctly resamples 14% of the time, while in parallel only 3% of the time. Conversely, serial confounds (as above) only 9% of the time while parallel confounds 28% of the time. Nevertheless, the problem is greatly mitigated when $D \gg P$.

Our SLDA model admits this assumption. In each iteration, the correct, corpus-wide (global) φ is copied to all worker nodes. z is then resampled *locally*, updating φ on each node as if all other nodes were idle. At the end of the iteration, when all nodes have resampled the topic assignment z of each word in each document, a `reduce` operation is used to add upp all nodes' separate counts

[2] θ does not have this problem, because it is local to the documents and the worker nodes.

and compute the correct global φ again. Then the next iteration starts. These iterations are repeated according to the Gibbs iteration parameter, allowing z to "converge."

At the end, φ and θ for the window in time t are computed and output to file annotated with the window's date. θ is saved via Hadoop, much like the tweets are stored. Each vocabulary is also saved to file. The model (defined by z) is not immediately discarded afterwards; it is used to calculate the prior of the next model. Ultimately, these θ_t and φ_t can then be used to discover trends and topics, and perform an evaluation of the model's performance.

3.4. Evaluation

Perplexity and two other comparative evaluations of the model were performed to measure different aspects of the model's performance, which are detailed in the sections below. In each case, a simple keyword-based model was used as a control (baseline). All these methods were tested versus "known" geo-trends, in particular the peak dates for the various states' primary elections in the U.S. Tweets near these dates (within half a week) that feature election topics were assumed to express the associated U.S. state relatively often.

3.4.1. Perplexity

A typical evaluation of LDA-like models uses *perplexity* to compare models' and their parameters' performance. Perplexity, on an unseen test document set $D*$, is

$$\text{perplexity}(D*) = \exp\left(-\frac{\mathcal{L}(D*)}{\sum_{d=1}^{D*} N_d}\right) = \exp\left(-\frac{\sum_{d=1}^{D*} \log p(v_{dn}|\varphi, \alpha)}{\sum_{d=1}^{D*} N_d}\right)$$

where $\mathcal{L}$ is the *log-likelihood* of the corpus [27]. In other words, perplexity measures how likely it is that the test corpus was generated by the model, i.e., to what extent the model explains the test corpus. Lower perplexity is better (more accurate).

Perplexity evaluates the model objectively, and thus does not allow easy comparison with non-LDA models. It is of limited use, because it can correlate poorly with human-perceived cohesiveness of topics. Nevertheless, it is useful to compare against different LDA models with different parameters. Here we calculate perplexity via approximation (sampling) for all models in a time interval from which a median is derived and plotted versus topic count K for some data set. The unseen test set is simply a small bisection (about 100–1000 tweets per window) of the corpus.

3.4.2. Evaluation Types

Evaluations were either *implicit* or *explicit.* For implicit evaluations, the SLDA and keyword models estimated location prevalence by comparing tweets in a sample set to tweets in the training set. The sample set was created in two ways, either by manually *writing* tweets about the election, or by *selecting* tweets about the election from an unseen training set. All location words were erased from these sample tweets.

For explicit evaluations, a human evaluator helped guide the estimation. For the SLDA model, the evaluator inspected the topics' top words (with locations excluded so as to blind the evaluator from bias) and for each time window, selected the topic best corresponding to the election.[3] For the baseline keyword model, the evaluator wrote about 20 keywords (such as `primary`, `candidate`, etc.) to be about the election.

Location Prediction. When SLDA completes, θ is inferred for each tweet in the training set. The assumption $\theta \sim \mathrm{Dir}_K(\alpha)$ can then be used to infer topic proportions for *any* tweet, allowing the model to make testable predictions. For *implicit* evaluation, θ is inferred for tweets in the sample set. The model, for some window, then used $\theta \cdot \varphi$ to predict the most likely location word in the tweet. If

[3] Sometimes, SLDA captures a very small facet of a larger trend as a topic, and its φ may mislead a human evaluator. To protect against this sparsity, the evaluator was aided by ranking topics after $\sum \theta$, i.e., topic prevalence in each time window.

the predicted location matches the election location at some time, that prediction is marked as correct. The procedure is similar for *explicit* evaluation, except that θ is 1 for the human-chosen topic and 0 elsewhere. The confidences of these predictions, normalized over the 10 most likely locations, were also recorded.

Similarly, the baseline keyword model, for *implicit* evaluation, searched for tweets in the training corpus that maximized the number of matched words with a sample tweet. Location words were then ranked after prevalence and match success in these tweets, from which the model can calculate confidence and make a prediction, as before. For *explicit* evaluation, the search tries to match the human-provided keywords instead.

Time Prediction. The discussed procedure was also repeated over all time windows to find, for each location, the point in time when that location was most prevalent for some trend. For both models, implicitly and explicitly, the peak (maximum) date of prevalence for every location was found. This predicted date was then compared to the assumed peak location date for the election. If the predicted date was less than half a week (3.5 days) away from the assumed date, it was marked as *correct*. Statistics about the predicted dates' deviations from the assumed date were also recorded.

As a side-effect, the procedure was also used to plot location prevalence for the election over time. Since there are thousands of location mentions, a selection had to be made. In the first type of plot, called a *manual* plot, locations were selected from the assumed election locations within the whole time span. In the second type of plot, called an *automatic* plot, the model automatically selects the 15 most prevalent locations.

4. Results

This section presents visualizations of SLDA's output for various corpora and parameters. Tables 3 and 4 present a selection of topics in the form of their most frequent words, as discovered by SLDA on the **USTwiNews** corpus with $K = 20$, low α, β and 4-day time windows. The topics are shown at three different points in time, illustrating their evolution over time. Words in **bold** are recognized as locations, *italics* as persons, and underline as organizations.

14–18 Mar 2016	
Topic no. 1	
Word	Prob. [%]
primary	7.71
win	3.64
day	3.00
#PrimaryDay	2.71
ohio	2.57
florida	2.02
2016	1.73
watch	1.47
tuesday	1.44
donald trump	1.44
campaign	1.44
gop	1.41

⇒

17–21 Mar 2016	
Topic no. 1	
Word	Prob. [%]
race	3.15
convention	2.40
win	1.51
donald trump	1.39
presidential	1.39
candidate	1.39
voter	1.39
gop	1.26
nomination	1.14
bernie sanders	1.14
hillary clinton	1.14
latest	1.01

⇒

20–24 Mar 2016	
Topic no. 1	
Word	Prob. [%]
tuesday	1.75
tytlive	1.70
western	1.51
2016	1.46
vote	1.46
#FinalFive	1.36
final	1.31
win	1.26
tonight	1.26
clinton	1.26
race	1.26
trump	1.22

Table 3: Most frequent words from topic no. 1 as it evolves during six days. Note the relative stability of the topic subject which seems to be about the American primary elections and differs completely from topic no. 2, see Table 4. The topic order in a data set is decided by its popularity with the topics displayed in decreasing order. #FinalFive refers to the remaining five presidential candidates.

14–18 Mar 2016	
Topic no. 2	
Word	Prob. [%]
police	2.94
kill	2.37
attack	2.29
man	2.13
die	1.64
officer	1.64
suspect	1.64
brussels	1.23
shooting	1.23
report	1.15
charge	1.15
raid	1.15

⇒

17–21 Mar 2016	
Topic no. 2	
Word	Prob. [%]
suspect	5.52
paris	5.32
attack	4.18
salah abdeslam	3.10
police	2.32
capture	2.32
raid	2.06
terror	1.75
belgian	1.60
arrest	1.55
report	1.45
belgium	1.39

⇒

20–24 Mar 2016	
Topic no. 2	
Word	Prob. [%]
brussels	6.38
attack	6.23
#Brussels	4.11
suspect	3.85
airport	3.31
police	2.59
paris	1.98
bomber	1.80
belgian	1.62
suicide	1.30
brussels	1.22
terror	1.08

Table 4: Most frequent words from topic no. 2 as it evolves during six days. Note that **brussels** during 14–18 Mar 2016 refers to the manhunt taking place there, not the Brussels attacks which happened on March 22$^{\text{nd}}$.

Figure 4 correspondingly shows the most commonly featured topics in the news corpus over the time period, i.e., topics k with the highest $\sum_{d=1}^{D} \theta_{dk}$. Note that the topic titles have been written by a human after inspection and interpretation of the topics' top words—they would otherwise be difficult to present in this type of chart.

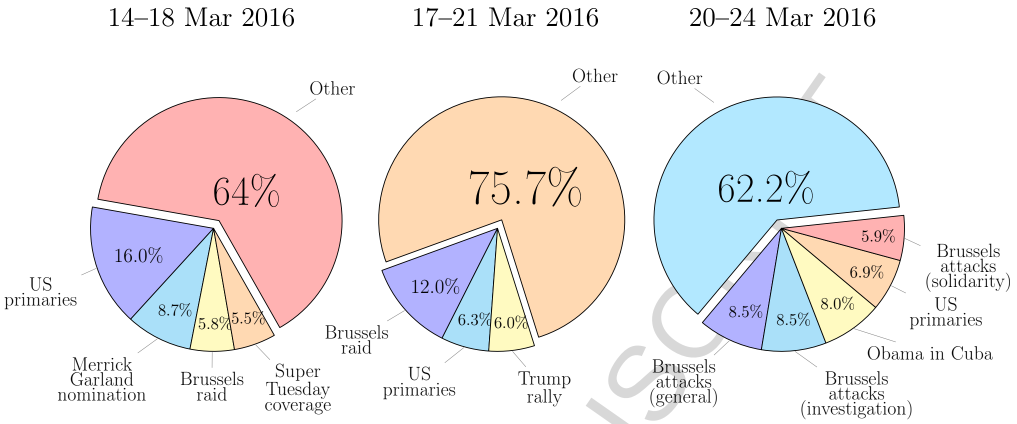


Figure 4: Proportions of featured topics in news documents for a 6-day time interval, i.e., $\sum \theta$, for the most frequent topics. Note that the topic titles are human-interpreted after inspection of the topics.

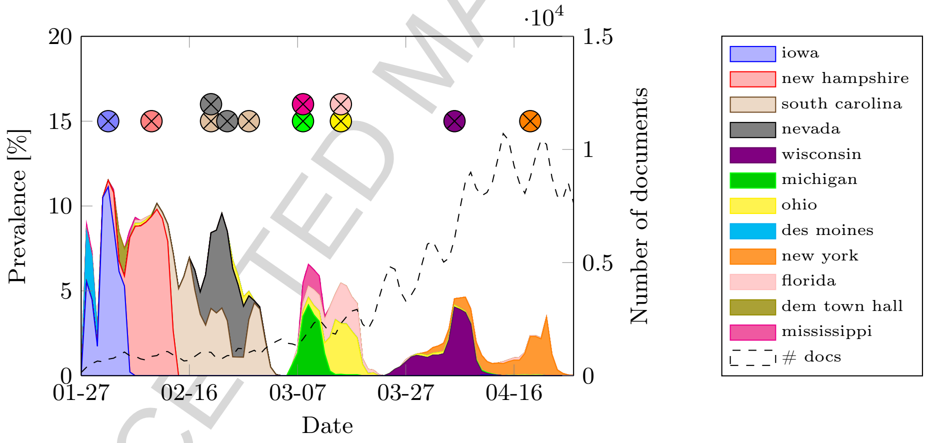


Figure 5: The 12 words identified as locations with the highest peak prevalences, as expressed by the human-identified election topic for each window. The round markers show actual election dates and locations. Note that locations are not filtered by state; hence *des moines* (Iowa capital) and *dem town hall* appear. The number of documents is drawn as a dashed line. Note that the anti-correlation between the peak heights and number of documents is likely due to φ becoming smoother as more location words appear.

As a first glance on location recognition, consider Figure 5. Here, an explicit evaluation of the **USTwiNews** corpus with $K = 20$, low α, β and 4-day time windows is shown. The human-identified topic concerning the American election over time is presented, with the prevalence of location words plotted over time. The prevalence is shown as accumulative proportions, and the locations have been selected automatically; that is, the 12 locations with the tallest peak prevalences are selected. Compare with the known election locations and dates shown with the markers. Note that locations are not filtered by state; hence *des moines* (Iowa capital) and *dem town hall* appear. The number of documents in each window is drawn as a dashed line. The anti-correlation between the peak heights and number of documents is likely due to φ becoming smoother as more location words appear.

5. Evaluation

This section presents the results from the evaluation as explained in Section 3.4.

5.1. Perplexity

Figure 6 shows perplexity as a function of the topic count K for various corpora and SLDA parameters. The perplexity shown is the median of perplexities for each window and calculated versus an unseen test set of withheld tweets. Figure 7 is a box plot showing how distributed the perplexities are as taken over time, for the USTwiNews corpus ($\alpha = 0.001$, $\beta = 0.0001$).

5.2. Location Prediction Keyword Model vs. SLDA

The baseline keyword model location prediction performance for the USTwiNews corpus (hashtags retained) is presented in Table 5 with evaluation types and measurement as in Section 3.4.2. A keyword evaluation of the USE2016 corpus has been omitted since as the “weaker” corpus, see Figure 6, it was judged to be comparatively unnecessary.

An SLDA model location prediction performance for the USTwiNews corpus is presented in Table 6 with evaluation types and measurement as in Section 3.4.2. The table shows performance for various values of K, with the other SLDA parameters being fixed at $\alpha = 0.001$, $\beta = 0.0001$, $c = 0.5$, window size $L = 4$, time slice 24 hours, 1000 Gibbs iterations and a vocabulary of the top 15,000 words. Hashtags are retained and split as in Section 3.2.

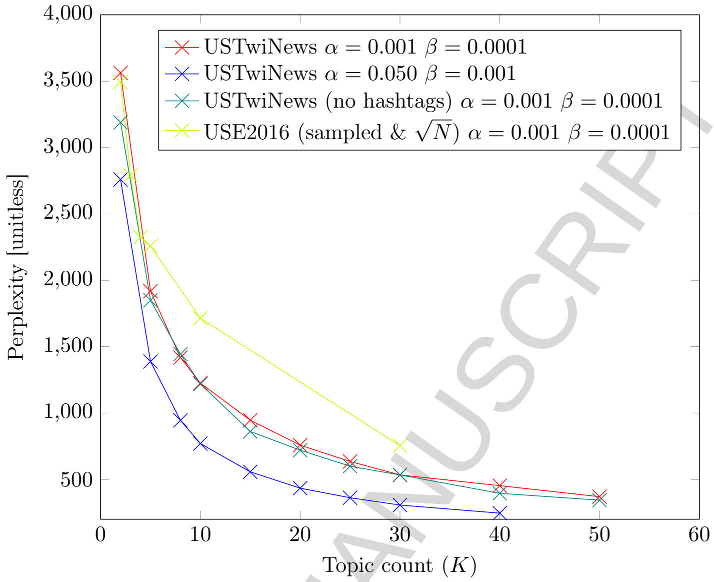


Figure 6: Perplexity as a function of topic count K. All plots have SLDA parameters $c = 0.5$, window size $L = 4$, time slice 24 hours, 1000 Gibbs iterations, and a vocabulary of the top 15,000 words.

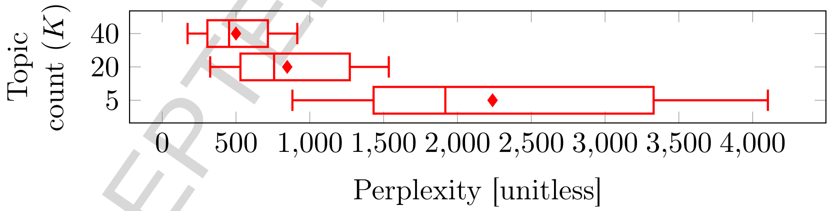


Figure 7: Box plot of perplexities for (USTwiNews, $\alpha = 0.001$, $\beta = 0.0001$) showing the distribution of perplexity samples (each sample is the perplexity measured on a particular time window).

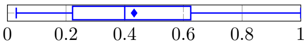
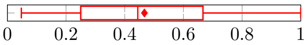
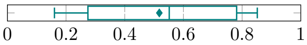
evaluation type	correct predictions	confidence
implicit (writing)	466/1515 (30.8%)	
implicit (selecting)	332/750 (44.3%)	
explicit	9/15 (60.0%)	

Table 5: Keyword model location prediction accuracy and confidence on the USTwiNews corpus for the evaluation types as described in Section 3.4.2. The confidences of the predictions, normalized over the ten most likely locations, have been visualized using box plots.

Another parametrization of the same corpus is shown in Table 7. Only the value of K resulting in the highest accuracy has been included. The other SLDA

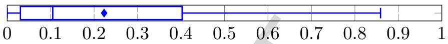
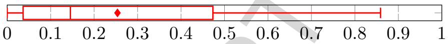
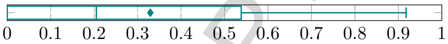
K	evaluation type	correct predictions	confidence
5	implicit (writing)	5510/15140 (36.4%)	
5	implicit (selecting)	3160/7500 (42.1%)	
20	explicit	8/15 (53.3%)	

Table 6: SLDA location prediction accuracy and confidence as a function of topic count K and the evaluation types as in Section 3.4.2 on the USTwiNews corpus—with hashtags **retained**. The SLDA parameters are $\alpha = 0.001$, $\beta = 0.0001$, $c = 0.5$, window size $L = 4$, time slice 24 hours, 1000 Gibbs iterations and a vocabulary of the top 15,000 words.

parameters are fixed at $\alpha = 0.05$, $\beta = 0.001$, $c = 0.5$, window size $L = 4$, time slice 24 hours, 1000 Gibbs iterations and a vocabulary of the top 15,000 words. Hashtags are retained and split as in Section 3.2.

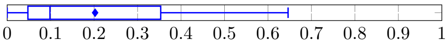
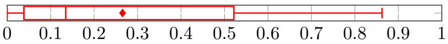
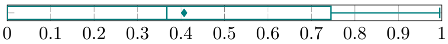
K	evaluation type	correct predictions	confidence
2	implicit (writing)	4288/15140 (28.3%)	
5	implicit (selecting)	3150/7500 (42.0%)	
30	explicit	9/15 (60.0%)	

Table 7: SLDA location prediction accuracy and confidence as a function of topic count K and the evaluation types as in Section 3.4.2 on the USTwiNews corpus—with hashtags **retained**. The SLDA parameters are $\alpha = 0.05$, $\beta = 0.001$, $c = 0.5$, window size $L = 4$, time slice 24 hours, 1000 Gibbs iterations and a vocabulary of the top 15,000 words.

The first parametrization was repeated for the USTwiNews corpus but with hashtags removed (and not split). The results for this parametrization are shown in Table 8.

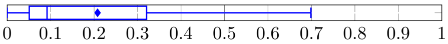
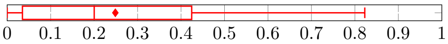
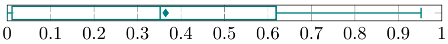
K	evaluation type	correct predictions	confidence
2	implicit (writing)	5981/15130 (39.5%)	
5	implicit (selecting)	3450/7500 (46.0%)	
20	explicit	9/15 (60.0%)	

Table 8: SLDA location prediction accuracy and confidence as a function of topic count K and the evaluation types as in Section 3.4.2 on the USTwiNews corpus—with hashtags **removed**. The SLDA parameters are $\alpha = 0.001$, $\beta = 0.0001$, $c = 0.5$, window size $L = 4$, time slice 24 hours, 1000 Gibbs iterations and a vocabulary of the top 15,000 words.

Finally, the location prediction performance for the USE2016 corpus is presented in Table 9. Hashtags are retained and split.

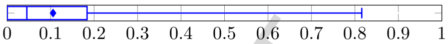
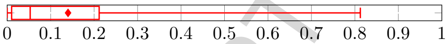
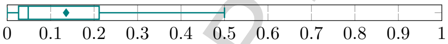
K	evaluation type	correct predictions	confidence
30	implicit (writing)	1304/11110 (11.7%)	
	implicit (selecting)	945/5500 (17.2%)	
	explicit	1/11 (9.1%)	

Table 9: SLDA location prediction accuracy and confidence as a function of topic count K and the evaluation types as in Section 3.4.2 on the USE2016 corpus—with hashtags **retained**. The SLDA parameters are $\alpha = 0.001$, $\beta = 0.0001$, $c = 0.5$, window size $L = 4$, time slice 24 hours, 1000 Gibbs iterations and a vocabulary of the top 15,000 words.

5.3. Time Prediction Keyword Model vs. SLDA

The baseline keyword model time prediction performance for the USTwiNews corpus is presented in Table 10 with evaluation types and measurement as in Section 3.4.2. Note that a keyword evaluation of the USE2016 corpus is omitted, because, as part of the selection of the "better" corpus, it was judged to be comparatively unnecessary.

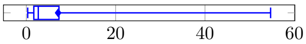
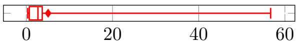
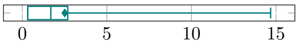
evaluation type	correct predictions	deviation in days
implicit (writing)	662/1515 (43.7%)	
implicit (selecting)	443/750 (59.1%)	
explicit	12/15 (80.0%)	

Table 10: Keyword model time prediction accuracy and deviation in days after the evaluation types as in Section 3.4.2 on the USTwiNews corpus.

For SLDA only the parametrization with the best time prediction accuracy is presented, which was found, for both corpora, to be with the SLDA parameters $\alpha = 0.001$, $\beta = 0.0001$, $c = 0.5$, window size $L = 4$, time slice 24 hours, 1000 Gibbs iterations, and a vocabulary of the top 15,000 words. These are presented as a function of K and evaluation type in Table 11 for USTwiNews and in Table 12 for USE2016.

Figure 8 (a) plots the explicitly evaluated keyword model and the reference locations over time as accumulative prevalences (word probabilities). The markers indicate these reference locations and their dates; some locations are summarized by nicknames for those dates, as there are too many to display in one label.

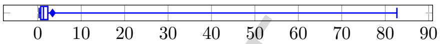
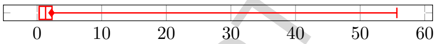
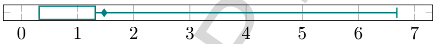
K	evaluation type	correct predictions	deviation in days
5	implicit (writing)	12905/15150 (85.2%)	
	implicit (selecting)	6728/7500 (89.7%)	
	explicit	14/15 (93.3%)	

Table 11: SLDA time prediction accuracy and confidence as a function of topic count K and the evaluation types as in Section 3.4.2 on the USTwiNews corpus. The SLDA parameters are $\alpha = 0.001$, $\beta = 0.0001$, $c = 0.5$, window size $L = 4$, time slice 24 hours, 1000 Gibbs iterations and a vocabulary of the top 15,000 words.

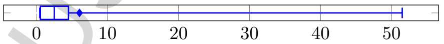
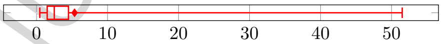
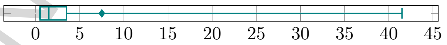
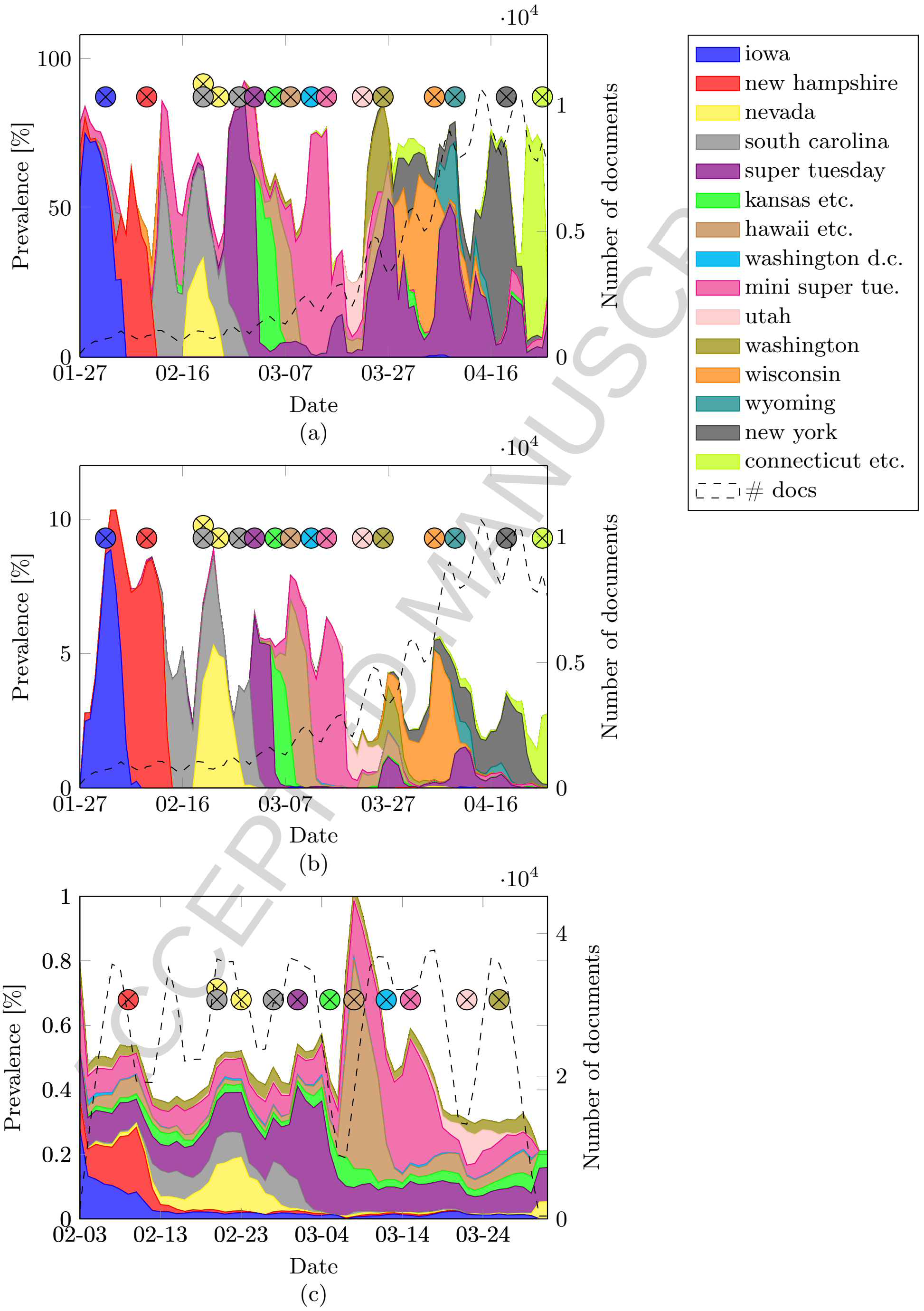
K	evaluation type	correct predictions	deviation in days
10	implicit (writing)	6258/11110 (56.3%)	
	implicit (selecting)	3130/5500 (56.9%)	
	explicit	8/11 (72.7%)	

Table 12: SLDA time prediction accuracy and confidence as a function of topic count K and the evaluation types as in Section 3.4.2 on the USE2016 corpus. The SLDA parameters are $\alpha = 0.001$, $\beta = 0.0001$, $c = 0.5$, window size $L = 4$, time slice 24 hours, 1000 Gibbs iterations and a vocabulary of the top 15,000 words.

Figure 8 (b) plots the USTwiNews human-chosen topic (explicit evaluation) and the reference locations over time as accumulative prevalences (word probabilities). The markers indicate these reference locations and their dates; some locations are summarized by nicknames for those dates, as there are too many to display in one label. Similarly, Figure 8 (c) plots the same, but for the USE2016 corpus.



(a)

(b)

(c)

Figure 8: The 15 location references and their prevalences as expressed by (a) tweets (in the USTwiNews corpus, hashtags retained) sharing as many human-selected election **keywords** as possible over time, (b) the human-identified election **topic** for each window in the USTwiNews corpus (hashtags retained), and (c) the human-identified election **topic** for each window in the USE2016 corpus (hashtags retained). For comparison, the known election locations and dates are shown with “⊗” markers. Some references (which have multiple and often very many locations) are summarized by nicknames. The number of documents in each window is drawn as a dashed line.

6. Discussion

The shortness of tweets limits their ability to convey news to approximately headline level. This is a barrier to unsupervised co-occurrence learning, because the same news may be conveyed using slightly different words and synonyms. This reduces the number of common words between equitopical documents. Fortunately, this can be somewhat compensated for by processing corpora with more documents, but yet remains a significant obstacle.

In general, the USE2016 corpus was found to perform much worse in both types of prediction than the USTwiNews corpus. The USE2016 corpus was created first, and after the poor results it was judged that another corpus was necessary for being able to perform a proper comparison. As such, keyword model performance is missing from this model.

The USTwiNews corpus was significantly more well-written and coherent than the USE2016 corpus. There are two obvious reasons for this. The first is that the USTwiNews tweets are written by professional news companies, and so employ both succinct and non-noisy language. Each such tweet has much higher average text quality than tweets written by the public, which helps learning. Secondly, the USTwiNews corpus is consistent and contains almost exclusively news, whilst the USE2016 corpus was created by capturing tweets using keywords which often results in wildly varying content.

6.1. Preprocessing

The preprocessor exceeded expectations upon examination of its bag-of-words output. The Stanford NLP lemmatizer and named entity recognizer were both very accurate and rarely made mistakes. Rarely, the NER would incorrectly classify words. For example, the word *zika* in *zika virus* was often classified as a person (it is a forest) and rarely long chains of words would be annotated as something congruent when they should be separate. Even more rarely, the lemmatizer would make mistakes, such as not lemmatizing verbs whose noun form is confoundable with the present participle (for example, “a *timing* issue”).

Curiously, preprocessing the hashtags did not appear to reduce the model’s perplexity. This may be due to that words not in the model’s vocabulary do not contribute to the perplexity, and so there is little effective difference. Manual inspection of the topics, however, reveals that preprocessing did make the topics more cohesive. This is likely because many tweets write an easily recognized phrase, for example `#BrusselAttacks`, as a hashtag. Deleting all hashtags removes vital information, both in the form of the hashtag itself and its possible text elements. As such, splitting hashtags and treating the result as words was deemed valuable.

Further inspection also revealed that persons are often referred in tweets by their twitter handle (for example, “`I like @realDonaldTrump more`”). Mentions were deleted in the preprocessing step entirely, losing information. On the other hand, correctly parsing such mentions can be a difficult NLP problem and is also left as an open question.

6.2. SLDA

As the illustrative results indicate, the SLDA model is very capable of automatically (unsupervisedly) capturing and categorizing tweets into coherent topics. Both estimation of φ and inference with θ shows that the model not only manages to understand the overall discussed subject during some time period, but also understand what subjects are being discussed in a given tweet. The USTwiNews data set seems best expressed with K between 15 and 20 topics.

In addition, the SLDA model as defined by Lau et al. works very well in updating the current categorization as new documents arrive [33]. The model is good at migrating current topics to fit new information, and replace topics that are fading in current trends.

Figure 5 clearly shows that the model is able to not only deduce the trending locations, but also match these fairly accurately to reference locations and dates. It should be noted, however, that prevalence (i.e., word probability) is a relative unit and says little about the popularity of a location without further interpretation.

Nevertheless, the SLDA model can overcome these problems by letting a user search for documents with high proportion of some chosen topic. As such, a user can then inspect the documents and perhaps better understand how the locations are being referred to. Such search capabilities makes SLDA very capable at tracking geographical trends.

An intrinsic shortcoming of the SLDA model is its inability to keep similar clusters assigned to the same topic label on larger time scales. Topics can swap locations and change meaning radically over time, as they become forced to represent the current subcorpus. In addition, the fixed K can become problematic if the corpus has strong structural changes over time. The former problem can possibly be overcome by augmenting the SLDA model with "supertopics"—predefined distributions over words which can be compared to deduced φ and labeled in an extra step.

6.3. Perplexity

SLDA yields decreasing perplexity with growing topic count K which is expected and well-known in the literature [27]. There is often a global minimum but one cannot be seen here. The decreasing perplexity is obviously due to that SLDA becomes armed with more topics to choose from when inferring a document. This naturally leads to a higher likelihood and thus a lower perplexity. Further, higher α and β give lower perplexity which is also expected. This yields less sparse distributions and empowers SLDA with much easier ways to "explain" odd words out, hence increasing likelihood.

6.4. SLDA prediction performance

When compared with the baseline keyword model, an optimal configuration of parameters of SLDA was found to have quite similar prediction accuracy. Further, explicit evaluation readily returns much better precision for both, which is both fair and expected. If a human can guide the models to calculate location proportions from explicit indications (keywords and topics respectively) instead of inferred ones, better precision is very natural. Moreover, human guidance is very natural for the case of LDA, as topics are usually inspected by humans anyway in most applications.

SLDA ($\alpha = 0.05$, $\beta = 0.001$) had a similar location prediction accuracy (but lower confidence) and a higher time prediction accuracy. The latter may be, as can be seen in Figure 8 (a), due to the additional peaks of Super Tuesday states. They have similar day deviance, however. Nevertheless, SLDA does not manage to greatly outperform the keyword-based model in precision which may have several explanations. One is that keywords are more "crisp" than topics, as φ does tend to be rather smooth. As such, SLDA is inherently less confident because of its less sparse probabilities. However, this may be an unfair judgment of LDA, because it manages to capture topics *automatically* and has no need to be "seeded" with keywords. It could perhaps then not be expected to perform as well because of its unsupervised nature.

Secondly, the corpora, as noted earlier, consist of tweets of short length and their bag-of-words representations may not be strong enough. This may also explain the relatively low ($\sim$ 50–60% in ideal conditions) precision of trends in both cases. It may be necessary to develop the model with advanced NLP to understand how the locations are being used in the text. However, not only is this a very difficult problem, it is also not obvious how to integrate this with typical formulations of LDA.

Both models have good precision and median deviance when it comes to predicting the *time* a location was most prevalent (with SLDA slightly better). This is also obvious from glancing at the plots. One could argue, however, that the *real* goal is to understand what location a topic is about at some time, not the other way around. Nevertheless, this result shows that the models are clearly resourceful and may simply suffer from limited understanding of the context that location words are placed in, due to the bag-of-words representation.

However, it should be stressed that LDA is a much better model for discovering topics than keyword-based models are, as the past literature has also discovered [27]. It is also more flexible and natural to perform searches, inferences and estimations with it, which are not obviously straightforward to do well with keyword models. From this the conclusion can be drawn that SLDA is a strong candidate for tracking geographical trends in streaming media, and is useful to gain oversights of topics and their most prevalent locations.

7. Conclusions

The ubiquity and increasing capability of social media have made it a growing target for study and extraction of useful information. To survey the massive amounts of information, automated software solutions is becoming an indispensable tool for being able to dig out and make sense of the data. In this paper, a prototypical tool is provided with which the topics and trending locations of streaming media can be automatically discovered. The tool provides key abilities to users to gain an overview of online discussions and features to search, rank, estimate, and infer documents and topics. The evaluation of this tool has granted a meaningful measure of its capability and illustration of its action.

In the work presented herein, the streaming latent Dirichlet allocation (SLDA) topic model has been evaluated to be an effective model for tracking geographical trends and topical locations. Conditional random fields were found to be a partial solution to the problem of geographical disambiguation, whereby good precision was obtained in recognition of ambiguous location terms, such as “washington,” by evaluation of their textual context. However, disambiguation of different places with the same name still remains. A possible solution could be to use the topical context that a location term appears in as described by the SLDA parameter φ, as a means to resolve such ambiguities. Whether this is possible to fully automate remains to answer, but prospects thus far appear good.

The research on topic models is still ongoing, and parallel research to its application to location correlation and discovery is relevant. There are many questions left open as to possible improvements, especially concerning creating a deeper understanding of the context that location words appear in.

Acknowledgments

This paper and research was supported by the European Union Seventh Framework Programme (grant agreement no. 312649).

References

[1] J. Brynielsson, F. Johansson, C. Jonsson, A. Westling, Emotion classification of social media posts for estimating people's reactions to communicated alert messages during crises, Security Informatics 3 (1) (2014) 1–11.

[2] D. Blei, Probabilistic topic models, Communications of the ACM 55 (4) (2012) 77–84.

[3] A. Sadilek, H. Kautz, J. P. Bigham, Finding your friends and following them to where you are, in: Proceedings of the fifth ACM international conference on Web search and data mining, ACM, 2012, pp. 723–732.

[4] L. Hong, A. Ahmed, S. Gurumurthy, A. J. Smola, K. Tsioutsiouliklis, Discovering geographical topics in the Twitter stream, in: Proceedings of the 21st international conference on World Wide Web, ACM, 2012, pp. 769–778.

[5] S. Abrol, L. Khan, F. B. Bastani, W. Wu, Location mining in online social networks, University of Texas, Dallas.

[6] B. Han, P. Cook, T. Baldwin, Text-based Twitter user geolocation prediction, Journal of Artificial Intelligence Research 49 (2014) 451–500.

[7] A. Schulz, A. Hadjakos, H. Paulheim, J. Nachtwey, M. Mühlhäuser, A multi-indicator approach for geolocalization of tweets, in: ICWSM, 2013, pp. 573–582.

[8] R. Li, S. Wang, H. Deng, R. Wang, K. C.-C. Chang, Towards social user profiling: unified and discriminative influence model for inferring home locations, in: Proceedings of the 18th

ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2012, pp. 1023–1031.

[9] M. Graham, S. A. Hale, D. Gaffney, Where in the world are you? Geolocation and language identification in Twitter, The Professional Geographer 66 (4) (2014) 568–578.

[10] R. Priedhorsky, A. Culotta, S. Y. Del Valle, Inferring the origin locations of tweets with quantitative confidence, in: Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing, ACM, 2014, pp. 1523–1536.

[11] A. Ahmed, L. Hong, A. J. Smola, Hierarchical geographical modeling of user locations from social media posts, in: Proceedings of the 22nd international conference on World Wide Web, ACM, 2013, pp. 25–36.

[12] C. Budak, T. Georgiou, D. Agrawal, A. El Abbadi, Geoscope: Online detection of geo-correlated information trends in social networks, Proceedings of the VLDB Endowment 7 (4) (2013) 229–240.

[13] C. Wang, J. Wang, X. Xie, W.-Y. Ma, Mining geographic knowledge using location aware topic model, in: Proceedings of the 4th ACM workshop on Geographical information retrieval, ACM, 2007, pp. 65–70.

[14] F. Wang, H. Wang, K. Xu, R. Raymond, J. Chon, S. Fuller, A. Debruyn, Regional level influenza study with geo-tagged Twitter data, Journal of Medical Systems 40 (8) (2016) 1–8.

[15] M.-H. Tsou, J.-A. Yang, D. Lusher, S. Han, B. Spitzberg, J. M. Gawron, D. Gupta, L. An, Mapping social activities and concepts with social media (Twitter) and web search engines (Yahoo and Bing): a case study in 2012 US presidential election, Cartography and Geographic Information Science 40 (4) (2013) 337–348.

[16] J. Kazama, K. Torisawa, Exploiting Wikipedia as external knowledge for named entity recognition, in: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), 2007, pp. 698–707.

[17] A. E. Richman, P. Schone, Mining wiki resources for multilingual named entity recognition, in: ACL, 2008, pp. 1–9.

[18] A. Ritter, S. Clark, O. Etzioni, Named entity recognition in tweets: An experimental study, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2011, pp. 1524–1534.

[19] L. Chiticariu, R. Krishnamurthy, Y. Li, F. Reiss, S. Vaithyanathan, Domain adaptation of rule-based annotators for named-entity recognition tasks, in: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2010, pp. 1002–1012.

[20] A. Mikheev, M. Moens, C. Grover, Named entity recognition without gazetteers, in: Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 1999, pp. 1–8.

[21] S. Cucerzan, D. Yarowsky, Language independent named entity recognition combining morphological and contextual evidence, in: Proceedings of the 1999 Joint SIGDAT Conference on EMNLP and VLC, 1999, pp. 90–99.

[22] W. Zhang, J. Gelernter, Geocoding location expressions in Twitter messages: A preference learning method, Journal of Spatial Information Science 2014 (9) (2014) 37–70.

[23] J. R. Finkel, T. Grenager, C. Manning, Incorporating non-local information into information extraction systems by Gibbs sampling, in: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 2005, pp. 363–370.

[24] T. Hofmann, Probabilistic latent semantic indexing, in: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 1999, pp. 50–57.

[25] S. Shivashankar, S. Srivathsan, B. Ravindran, A. V. Tendulkar, Multi-view methods for protein structure comparison using latent Dirichlet allocation, Bioinformatics 27 (13) (2011) i61–i68.

[26] J. Liang, P. Liu, J. Tan, S. Bai, Sentiment classification based on AS-LDA model, Procedia Computer Science 31 (2014) 511–516.

[27] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent Dirichlet allocation, the Journal of machine Learning research 3 (2003) 993–1022.

[28] Y. W. Teh, M. I. Jordan, M. J. Beal, D. M. Blei, Hierarchical Dirichlet processes, Journal of the american statistical association.

[29] C. Wang, J. Paisley, D. Blei, Online variational inference for the hierarchical dirichlet process, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 2011, pp. 752–760.

[30] C. Chemudugunta, P. S. M. Steyvers, Modeling general and specific aspects of documents with a probabilistic topic model, in: Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference, Vol. 19, MIT Press, 2007, p. 241.

[31] L. AlSumait, D. Barbará, C. Domeniconi, On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking, in: Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on, IEEE, 2008, pp. 3–12.

[32] M. Hoffman, F. R. Bach, D. M. Blei, Online learning for latent Dirichlet allocation, in: advances in neural information processing systems, 2010, pp. 856–864.

[33] J. H. Lau, N. Collier, T. Baldwin, On-line trend analysis with topic models: #twitter

trends detection topic model online, in: COLING, 2012, pp. 1519–1534.

[34] L. Wan, L. Zhu, R. Fergus, A hybrid neural network-latent topic model, in: International Conference on Artificial Intelligence and Statistics, 2012, pp. 1287–1294.

[35] Z. Cao, S. Li, Y. Liu, W. Li, H. Ji, A novel neural topic model and its supervised extension, in: AAAI, 2015, pp. 2210–2216.

[36] P. Xie, D. Yang, E. P. Xing, Incorporating word correlation knowledge into topic modeling, in: HLT-NAACL, 2015, pp. 725–734.

[37] T. L. Griffiths, M. Steyvers, Finding scientific topics, Proceedings of the National Academy of Sciences 101 (suppl 1) (2004) 5228–5235.

[38] X. Wang, E. Grimson, Spatial latent Dirichlet allocation, in: Proceedings of the 20th International Conference on Neural Information Processing Systems, Curran Associates Inc., 2007, pp. 1577–1584.

[39] Twitter, REST APIs, `https://dev.twitter.com/rest/public`, accessed: April 19 (2016).

[40] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, I. Stoica, Spark: Cluster computing with working sets, HotCloud 10 (10-10) (2010) 95.

[41] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, D. McClosky, The Stanford CoreNLP natural language processing toolkit, in: Association for Computational Linguistics (ACL) System Demonstrations, 2014, pp. 55–60.
URL `http://www.aclweb.org/anthology/P/P14/P14-5010`

[42] Stop words, `http://www.nada.kth.se/~markusos/data/stopwords.txt`, accessed: April 20 (2016).

[43] D. Newman, A. Asuncion, P. Smyth, M. Welling, Distributed algorithms for topic models, The Journal of Machine Learning Research 10 (2009) 1801–1828.

Biography

Marianela García Lozano is a senior scientist at the Swedish Defence Research Agency (FOI) since 2001, working with information and knowledge modelling, modelling and simulation, software development in distributed systems, semantic web, web mining and big data. She has managed several projects within the domain of semantic information management, distributed systems and sensor services since 2004 and has also been project leader in several EU funded projects. Marianela received her M.Sc. degree in Computer Science in 2003 and her Licentiate degree in Electronic and Computer Systems in 2010 from the Royal Institute of Technology (KTH). Marianela is currently pursuing her Ph.D.

Jonah Schreiber works in the distributed systems group at Google, Mountain View, California. Jonah holds an M.Sc. in Engineering Physics (2016) from the Royal Institute of Technology (KTH) in Stockholm, Sweden. His research interests include machine learning, natural language processing, and visualization.

Joel Brynielsson is a deputy research director at the Swedish Defence Research Agency (FOI) and an associate professor at the Royal Institute of Technology (KTH). Joel is Docent (Habilitation) in Computer Science (2015), and holds a Ph.D. in Computer Science (2006) and an M.Sc. in Computer Science and Engineering (2000) from KTH. His research interests include uncertainty management, information fusion, probabilistic expert systems, the theory and practice of decision-making, game theory, web mining, privacy-preserving data mining, cyber security, and computer security education.

Highlights

- Online discussions can be tracked geographically over time
- A distributed geo-aware streaming LDA (SLDA) model has been developed
- Evaluation was performed during the 2016 American presidential primary elections
- It was shown that locations correlated with the actual election locations
- SLDA was shown to be an effective model for tracking topics and geographical trends