WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS
DOI: 10.37394/23209.2021.18.13

Examining LDA2Vec and Tweet Pooling for Topic Modeling on Twitter Data

Kristofferson Culmer & Jeffrey Uhlmann
College of Engineering
University of Missouri-Columbia
Columbia, Missouri
USA

Abstract: The short lengths of tweets present a challenge for topic modeling to extend beyond what is provided explicitly from hashtag information. This is particularly true for LDA-based methods because the amount of information available from per-tweet statistical analysis is severely limited. In this paper we present LDA2Vec paired with temporal tweet pooling (LDA2Vec-TTP) and assess its performance on this problem relative to traditional LDA and to Biterm Topic Model (Biterm), which was developed specifically for topic modeling on short text documents. We paired each of the three topic modeling algorithms with three tweet pooling schemes: no pooling, author-based pooling, and temporal pooling. We then conducted topic modeling on two Twitter datasets using each of the algorithms and the tweet pooling schemes. Our results on the largest dataset suggest that LDA2Vec-TTP can produce higher coherence scores and more logically coherent and interpretable topics.

Key-Words: Topic Modeling, NLP, Twitter, Tweet Pooling, LDA, LDA2Vec, Biterm

Received: January 24, 2021. Revised: July 2, 2021. Accepted: July 17, 2021. Published: July 31, 2021.

1 Introduction

The rapid growth of social media over the past twenty-plus years has led to massive amounts of digital information being curated online. Of note, a large social media platform like Twitter, which reports over 330 million average monthly active users, allows its users to make posts, called *tweets*, that have a maximum text length of 280 characters. In 2020, Twitter reported an estimated 500 million daily tweets on their platform. As a result of such a large corpus, Twitter provides a veritable treasure trove for Natural Language Processing (NLP) researchers [1]. One of the challenges facing NLP researchers in mining large corpora is discovering the underlying topics and themes within the text. Various approaches to topic modeling have been developed to address this challenge.

Topic modeling identifies latent patterns of word occurrence using the distribution of words in a collection of documents. The output is a set of topics consisting of clusters of words that co-occur in these documents according to certain patterns [2] [3] [4]. Topic modeling has been used by researchers in a number of fields to analyze text corpora because of the opportunity it presents to gain a deeper understanding of the human thought, sentiment, and opinions present in text [5]. Topic modeling has been used to classify Twitter trends [6], large-scale event detection on Twitter [7] [8], analyze public perception of COVID19 [9], analyze articles in traditional media and compare them to Twitter [10], identify topics in political speech [11], mine information from software repositories and extract topics from source code [12], and identify depression-related language in Twitter [13].

One of the uses of Twitter, along with other social networks that have increased in popularity in recent years, has been to coordinate activist campaigns. This is called *social justice activism*, or hashtag activism; a term coined by media outlets which refers to the use of Twitter's hashtags for internet activism. The term can also be used to refer to the act of showing support for a cause through a *like*, *share/retweet*, etc., on any social media platform, such as Facebook or Twitter[14]. Recent popular hashtag activist campaigns have been #BlackLivesMatter, #MeToo, #WomensMarch, and #MentalHeath. Increasing racial tensions in the United States in recent years has brought more focus to analyzing tweets that use #BlackLivesMatter to gain better understanding of the movement and perceptions of it. As the #BlackLivesMatter hashtag increased in popularity what seemed as a competing hashtag, #AllLivesMatter began to appear and be used in a way that appeared to attempt to invalidate #BlackLivesMatter, which has caused increased interest in analyzing these tweets. Topic modeling on tweets, however, presents its own challenges.

Since 2018 tweets have been limited to 280 characters, and prior to that the limit was 140 characters. Because of the shortness of tweets, foundational topic modeling algorithms like Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA) and many of its variant algorithms like Online LDA

(OLDA), Author-Topic Model (ATM), and LDA2Vec do not perform well on tweet analysis because of sparsity in the term-document matrix [2] [3]. To address this issue a number of topic modeling algorithms like Twitter LDA, Temporal LDA (TM-LDA), and Biterm Model (BT) have been developed specifically for analyzing short texts like tweets. Some algorithms have proposed aggregating tweets based on various criteria: author, time, or conversation in an attempt to lengthen the tweet documents.

In this paper we propose a novel approach to topic modeling on tweets by using LDA2Vec paired with temporal tweet pooling, which we refer to as LDA2Vec-TTP. LDA2Vec enhances LDA by adding word context from word embeddings produced using Word2Vec, which yields good results on longer text documents but suffers from the same sparsity problem as LDA. We enhance LDA2Vec by aggregating tweets temporally and then compare our approach to two existing topic modeling algorithms. We carry out our experiments on #BlackLivesMatter and #AllLivesMatter tweet datasets, with tweets spanning more than 12 months, to gain a better long term understanding of these two popular hashtags on Twitter.

The format of the remainder of this paper is as follows: In section 2 we review and discuss related work in this domain of research. We cover classic topic modeling algorithms and also discuss their variants. We also discuss algorithms developed specifically for short texts and Twitter and how to evaluate topic models. In section 3 we present LDA2Vec and temporal pooling (LDA2Vec-TTP). In section 4 we present the experiment design and our methods. In section 5 we present the results of the experiment. In section 6 we summarize the results of our experiment and propose future directions for this research.

2 Related Work

In this section, we focus on the technical background and the evolution of topic modeling which will set the foundation that is required prior to diving deeper into our proposed method. We will discuss the different types of topic modeling, challenges in the field, recent trends in research, and the applications of topic modeling.

2.1 Approaches to Topic Modeling

There are two broad approaches to topic modeling: supervised and unsupervised. In supervised topic modeling the topics are predetermined and labeled data is used to train a model to classify unseen documents as belonging to a particular topic or multiple topics. In unsupervised topic modeling the topics are not known beforehand. The models are developed and trained to discover statistically significant words within documents and across the corpus. A topic consists of a collection of words and a human is needed to interpret the meaning of the topic. In this study we focus primarily on unsupervised topic modeling methods.

2.1.1 Topic Modeling Algorithms

Topic models are probabilistic statistical models that uncover the hidden thematic structure in document collections and provides a simple way to analyze large volumes of unlabeled text. The primary goal of the topic modeling is to uncover patterns of words in text and discover hidden structural words that runs through corpus by analyzing different patterns present in documents. [1]. We will now review some of the classic topic modeling algorithms.

2.1.2 Latent Semantic Analysis (LSA)

Latent Semantic Analysis (LSA) was proposed in 1998 as a fully automatic mathematical/statistical technique for extracting and inferring relations of expected contextual usage of words in passages of discourse. LSA is not a traditional natural language processing (NLP) or artificial intelligence (AI) program; as it uses no humanly constructed dictionaries, knowledge bases, semantic networks, grammars, syntactic parsers, or morphologies, or the like, and takes as its input only raw text parsed into words defined as unique character strings and separated into meaningful passages or samples such as sentences or paragraphs [15] [16].

The core idea behind LSA is its vector-based representation of hidden semantic context using Single Value Decomposition (SVD) to reduce the dimensions on its original matrix [1]. LSA has four main steps [17]:

1. Term-Document Matrix: a matrix is constructed where rows represent individual words and columns represent documents.

2. Transformed Term-Document Matrix: the values in the term-document matrix can be raw word counts but are often transformed. The best performance is observed when frequencies are cumulated using nonlinear values, such as the log of the frequency.

3. Dimension Reduction: SVD is used to reduce the dimensions of the matrix.

4. Retrieval in Reduced Space: Similarities are calculated among entries in the reduced dimensional space produced in step 3.

2.1.3 Probabilistic Latent Semantic Analysis (PLSA)

Probabilistic Latent Semantic Analysis (PLSA) improves on LSA by using a probabilistic model instead of SVD to detect topics [16].

2.1.4 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) is a generative probabilistic model of a corpus. LDA presumes that a corpus is characterized by a Dirichlet distribution. The basic idea behind this is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words [18]. We can, therefore, summarize a Dirichlet distribution as a distribution of distributions; in this case documents as a distribution of topics and topics as a distribution of words.

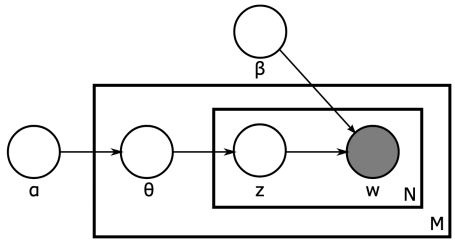

Figure 1: LDA Plate Diagram

In the figure 1 α and β are hyper parameters and must be set based on the data being processed. Correct values of α and β are necessary to produce a good model. The α term represents document-topic density. Large values correspond to more topics per document and conversely, smaller values correspond to fewer topics per document. The β term represents topic-word density. Large values correspond to more words per topic and conversely, smaller values correspond to fewer words per topic.

- α: document-topic density
- β: topic-word density
- M: number of documents in the corpus
- N: number of words in each document
- Θ: topic distribution for document M
- z: the topic assigned to each word w
- w: word in a topic

These classical approaches to topic modeling provide the basis for many other algorithms that extend their functionality. One of the deficiencies with LDA and LSA is that they treat each document as a bag of words (BOW), meaning that they neglect word order in the documents. This often time leads to unnatural word-topic assignment. [2]. Another issue with LDA and LSA is that they do not work well on short texts

2.1.5 Variants of Classic LDA and LSA

Authors in [19] [20] [21] address the BOW representation of documents by using n-gram representation of words where n-grams are a contiguous sequence of tokens in a document. The Author-Topic Model [22] extends LDA to include authorship information. Each author is associated with a multinomial distribution over topics and each topic is associated with a multinomial distribution over words. A document with multiple authors is modeled as a distribution over topics that is a mixture of the distributions associated with the authors.

In Online Latent Dirichlet Analysis (OLDA) the algorithm works in an online fashion which such that it incrementally builds an up-to-date model when a new document appears with no need to access previous information. [23]. Spherical topic modeling (STM) was proposed in [24] and uses term frequency - inverse document frequency (tf-idf) for feature extraction and models documents as points on in high-dimensional space, which allows for comparing document similarity using cosine distance.

There are other approaches which have used term-weighting as a means to improve LDA. [25] proposed Weighted Topic Model (WTM) and Balance Weighted Topic Model (BWTM) approaches for extracting the features in the corpus using IDF method. WTM concentrated on weighing all behaving words and resulting in the low parameter. Topic distributions of words increases its iterations. Efficiency was decreased in resulting high probability of topics. To achieve the efficiency BWTM used to manage more weights for specific words. Fewer weights are assigned for unspecific words. Term Weighting LDA (TWLDA) is proposed in [26] which assigns low weights to words with low topic discriminating power. Words with lower weights generally have weaker negative effect on results of LDA. This approach assigns topic-indiscriminate words low weights through a supervised term-weighting scheme to minimize their effect on topic assignment. The approach in [27] is similar to the approach in [26] as it also focused on weighting terms to minimize the effect of high-frequency background words that appear throughout the corpus, and using LDA as a baseline method.

2.1.6 Topic Modeling on Short Texts

As mentioned earlier, classic LDA does not work well on short text because conventional topic models implicitly capture the document-level word co-occurrence patterns to reveal topics, which leads to

severe data sparsity in short documents. [21]. The biterm topic model (BTM) [21] [28] addresses data sparsity by capturing co-occurrence by generating biterms. A biterm denotes an unordered word-pair co-occurring in a short context. In BTM the short context refers to a proper text window containing meaningful word co-occurrences. BTM views each short document as an individual context unit. Any two distinct words in a short text document is a biterm. For example, in the short text document "I visit apple store.", if we ignoring the stop word "I", there are three biterms, i.e. "visit apple", "visit store", "apple store".

The approach used in [29] uses a Gaussian Mixture Topic Model to construct word co-occurrence to address sparsity but different from [21] it is able to capture longer word contexts than biterms. In this approach, each word is projected into a vector that represents similarity between words within the contextual window. Hence, this approach can potentially capture context beyond unordered word-pair co-occurrences.

2.1.7 Topic Modeling in Twitter

Twitter tweets (documents) are inherently short as they are restricted in size by the Twitter platform. Tweet length had been restricted to 140 characters until 2018 when the platform doubled the maximum length to 280 characters. Aside from the topic modeling approaches developed specifically for short texts, here have been a algorithms developed specifically to conduct topic modeling in Twitter.

Twitter-LDA is proposed in [10] and it makes the assumption that each contains one topic only. It models the tweet generation process assuming that when writing a tweet, a user first chooses a topic based on her topic distribution. Then she chooses a bag of words one by one based on the chosen topic or the background model. This method qualitatively outperformed classic LDA in assigning topics to a set of tweets.

The Topic Tracking Model for Twitter (TTM) was developed as an improvement to Twitter-LDA in [30]. TTM improves on Twitter-LDA by modeling the dynamic nature of Twitter user's interests and topic trends changing. And unlike Twitter-LDA, TTM is an online algorithm and is able classify new tweets with having to build a new model.

2.2 Tweet Pooling

Algorithms have also been developed that employ document aggregation to address the short document problem. Document aggregation on tweets is called tweet pooling. The following list presents tweet pooling methods.

- **Author-based Pooling**: Tweets belonging to the same author are pooled together. A document for each author is built where all their tweets are combined [31] [28] [32].

- **Burst-Score pooling**: Trending topics on Twitter consists of one or more terms and a time period. In this scheme, bursts in term frequencies in a time window are calculated and tweets are pooled based on terms that experienced bursts. If a tweet contains multiple burst terms it is place in the pool for both those terms [31].

- **Temporal pooling**: Tweets that appear within the same hour are pooled together when a major event occurs on Twitter. Major events are detected and characterized by many tweets on the same topic appearing in a short period of time [31].

- **Hashtag-based Pooling**: Tweets containing the same hashtag are pooled together. 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 tweets containing the same hashtag can be pooled together[31].

- **Conversation-based pooling**: tweets and their replies are aggregated into a single document and the users who posted them are considered In this pooling scheme, co-authors in this scheme [33].

2.3 Evaluating Topic Models

The effectiveness of a topic modeling algorithm has typically been measured in one of three ways [34].

- **Preplexity**: calculates the likelihood the language model will correctly assign an unseen document to the correct topics [35].

- **Coherence**: topics are considered to be coherent if all or most of the words, in a the topic's top N words, are related [36].

- **Human interpretability**: measures the degree to which a human agrees with the topics assigned by a language model , the degree to which the collection of topics makes sense, and how well they associate with the documents and the corpus [34].

Although perplexity is useful for evaluating the language model's predictive power, coherence measures correlate more with semantically interpretable topics and more closely align with human judgement [34] [37].

3 LDA2Vec with Tweet Pooling

In this section we present our proposed approach of combining LDA2Vec with temporal tweet pooling (LDA2Vec-TTP). The motivation for this is to take advantage of the word contextual features of LDA2Vec while addressing the data sparsity problem. The algorithm architecture for LDA2Vec is shown in figure 2 and implements a hybrid approach to topic modeling, mixing the sparse document representations with dense word and topic vectors. The explanation of the LDA2Vec model is based on the work presented in [38].

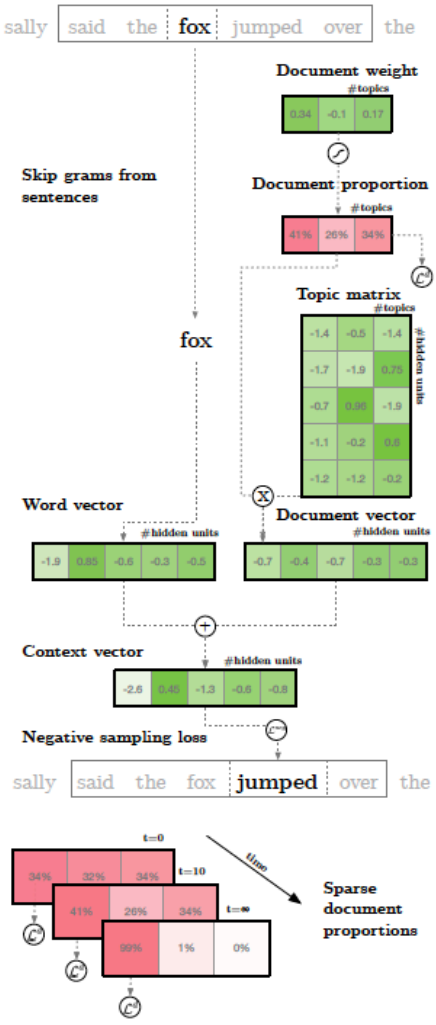

Figure 2: LDA2Vec Network Architecture

3.1 Word Representation

The SGNS loss shown in (3) attempts to discriminate context-word pairs that appear in the corpus from those randomly sampled from a 'negative' pool of words. This loss is minimized when the observed words are completely separated from the marginal distribution [38]. Pairs of pivot and target words (j, i) are extracted when they co-occur in a moving window scanning across the corpus. For every pivot-target pair of words the pivot word is used to predict the nearby target word. Each word is represented with a fixed length dense distributed-representation vector, where the same word vectors are used in both the pivot and target representations.

3.2 Document Representation

LDA2Vec embeds both words and document vectors into the same space and trains both representations simultaneously. By adding the pivot and document vectors together, both spaces are effectively joined. In LDA2Vec the context vector is explicitly designed to be the sum of a document vector and a word vector as in (1):

$$\overrightarrow{c_j} = \vec{w}_j + \vec{d}_j \tag{1}$$

3.3 Loss Function

The total loss term $\mathcal{L}$ in (2) is the sum of the Skipgram Negative Sampling Loss (SGNS) with the addition of a Dirichlet-likelihood term over document weights, $\mathcal{L}^d$ which is discussed in section 3.5. The loss is conducted using a context vector, c_j, pivot word vector w_j, target word vector w_i, and negatively-sampled word vector w_l.

$$\mathcal{L} = \mathcal{L}^d + \Sigma_{ij}\mathcal{L}_{ij}^{neg} \tag{2}$$

$$\mathcal{L}_{ij}^{neg} = \log\sigma\left(\overrightarrow{c_j}\cdot\vec{w}_i\right) + \Sigma_{l=0}^{n}\log\sigma\left(-\overrightarrow{c_j}\cdot\vec{w}_l\right) \tag{3}$$

3.4 Document Mixture

LDA2Vec generates a document vector from a mixture of topic vectors and to do so, we begin by constraining the document vector dj to project onto a set of latent topic vectors t_0; t_1,..., t_k. Each weight is a fraction that denotes the membership of document j in the topic k.

$$\vec{d}_j = p_{j0}\cdot\overrightarrow{t_0} + p_{j1}\cdot\overrightarrow{t_1} + \ldots + p_{jk}\cdot\overrightarrow{t_k} + \ldots + p_{jn}\cdot\overrightarrow{t_n} \tag{4}$$

3.5 Sparse Membership

The document weights p_{ij} are sparsified by optimizing the document weights with respect to a Dirichlet likelihood with a low concentration parameter α:

$$\mathcal{L}^d = \lambda\Sigma_{jk}(\alpha - 1)\log p_{jk} \tag{5}$$

The overall objective in (5) measures the likelihood of document j in topic k summed over all available documents. The strength of this term is modulated by the tuning parameter lambda. This simple

likelihood encourages the document proportions coupling in each topic to be sparse when alpha less than 1 and homogeneous when alpha greater than 1.

3.6 Temporal Pooling

LDA and its variants do not perform well on short text documents like tweets and so we consider the approaches used in [31] [28] [32] [33] that pool tweets based on various criteria.

Temporal pooling is used is to aggregate and analyze tweets when major events are detected[31]. While each tweet represents a unit of thought by the author, tweet's belonging to the same hashtag conversation provide context to the larger conversation at discrete points in time. LAD2Vec-TTP takes advantage of the context word vectors from LDA2Vec provide while temporal pooling aggregates tweets in discrete time windows. This approach allows us to more effectively analyze longstanding conversations on Twitter. We view the #BlackLivesMatter and #AllLivesMatter movements as longstanding, sustained events and it is, therefore, appropriate to use temporal pooling to aggregate tweets. We use a window of 1 day to aggregate the tweets in our datasets.

4 Experiment Design and Methods

Figure 3 shows the design for our experiment. We first collect tweets from the Twitter API and perform pre-processing on that data to prepare it to be analyzed. Next, we pool our #BlackLivesMatter and #AllLivesMatter datasets using three tweet pooling schemes: no pool, author-based pooling, and temporal pooling (1 day window). We then conduct topic modeling on each pool of tweets using LDA2Vec, Biterm topic model, and LDA. Finally, we compare the performance of each topic model and tweet pool pair using four coherence measures. The following sections explain in more detail the methods used at each stage of the experiment.

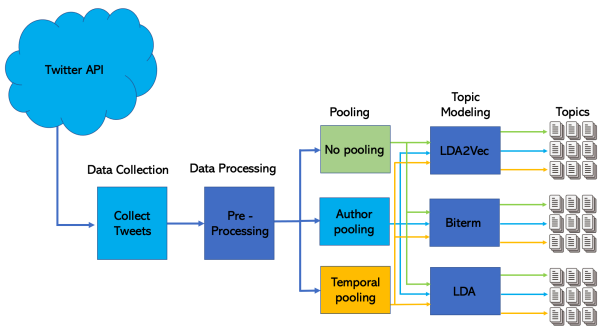

Figure 3: Experiment Design

4.1 Data Collection

We obtained tweet IDs in csv format from [39] for tweets containing #BlackLivesMatter or #AllLivesMatter (case-insensitive), collected over the time period from August 8th, 2014 to August 31st, 2015. Tweets were collected from the Twitter Gardenhose, which represents a 10% sample of all tweets. We focus on tweets from this time period because this is when the #BlackLivesMatter began being used prominently on Twitter. We also obtained a Twitter developer account which provides us with consumer keys and access tokens to access the Twitter API. We used the Twarc python module to access the API and download the tweets using the tweet IDs from the datasets. Tweets from the API are in json format.

#BlackLivesMatter	#AllLivesMatter
768582	101576

Table 1: Number of tweet IDs in each dataset

#AllLivesMatter Tweets

...How about #AllLivesMatter 👍
RT @abc4justice: #Justice4AlanBlueford #AllLivesMatter #EndPoliceTerror http://t.co/7zWDok7z0o
Stand for something or fall for anything ✊ #OccupyStamp #AllLivesMatter http://t.co/KqYuyDW0cG
This whole situation just saddens me! #AllLivesMatter 💕
....😆😆...I'm sleep though #blacklivesmatter #alllivesmatter #whiteprivilege #doesexist http://t.co/fvEJJnR5jg

#BlackLivesMatter Tweets

RT @taylorellison37: Crying for our society. 💔 #alllivesmatter http://t.co/e3BN4CkY2W
RT @Misz_ThickAuzzi: #AllLivesMatter #RIPMikeBrown 🙏❤️
"Happy Birthday, angel. #TamirRice 😔👦🏾💔 #BlackLivesMatter"
RT @Vlmani92: Today #TamirRice would have been 13 years old gone way too soon! Happy birthday #BlackLivesMatter 😢
RT @Leek_Mobb: America cares about #LoveWins but do you guys care about #BlackLivesMatter ? 💯✊🏾🟫

Figure 4: Tweets returned from the Twitter API

4.2 Data Preprocessing

Pre-processing is an essential step in NLP, text mining, and text classification because choosing the appropriate pre-processing tasks for the dataset will have an impact, positive or negative, on the results [40]. The tweets returned from the Twitter API need to be pre-processed before they can be used to train our topic modeling algorithms. Figure 4 shows a

sample of tweets from each dataset. The tweets contain urls, emojis, and other special characters and sequences like the ”RT” present in the tweets. RT in a tweet means it is a retweet and was not authored by the person who tweeted it. We look at the steps in pre-processing we carried out during this experiment in the next few subsections.

4.2.1 Removing Stop Words

Stop-words are the words that are commonly encountered in texts without dependency to a particular topic (conjunctions, prepositions, articles, etc.). Words such as ”is”, ”and”, ”at”, ”the”, and ”it” are considered English language stop words and do not contribute meaningfully to text classification. Therefore, the stop-words are usually assumed to be irrelevant in text classification studies, and removed prior to the classification. Stop-words are specific to the language being studied as in the case of stemming [40] [5].

We use the stop-word library from the nltk python module to remove the stop words from our tweets. We also remove #BlackLivesMatter and #AllLivesMatter hashtags from their respective corpora. Because those hashtags appear in every tweet of their respective corpus it will not provide any meaningful context to the topic modeling algorithms.

4.2.2 Removing Special Characters and Punctuation

As shown in figures 3 and 4 the tweets contain punctuation, standard emojis, and user created emojis from strings of special characters. We remove punctuation using the punctuation field in python’s string class. We use regular expressions to remove user-created and standard emojis. Figures 4 and 5 show a list of these emojis

":/", ":" , ":)" , ":-)" , ":(" , ":-/" , ":P"
":: ", "):" , "!!" , ">:(" , ":D" , ":-(" , ":@"

Figure 5: User-generated emojis removed during preprocessing

```
u"\U0001F600-\U0001F64F" #Emoticons
u"\U0001F300-\U0001F5FF" #Symbols & Pictographs
u"\U0001F680-\U0001F6FF" #Transport & Map symbols
u"\U0001F1E0-\U0001F1FF" #Flags (iOS)
u"\U00002702-\U000027B0"
u"\U000024C2-\U0001F251"
```

Figure 6: Standard emoji unicode ranges

4.2.3 Removing Duplicate Tweets

In Twitter a retweet occurs when a user another user’s tweet. We remove these retweets from our dataset because retweets represent an endorsement for the original tweet but do not represent a new thought or point of view. We, therefore, retain original tweets and remove all duplicate tweets.

4.2.4 Tokenization

Tokenization is the procedure of splitting a text into words, phrases, or other meaningful parts, namely tokens. In other words, tokenization is a form of text segmentation [40] [5]. All tweets in our dataset are tokenized.

4.2.5 Stemming and Lemmatization

The goal of stemming is to obtain root forms of derived words. For example, the words “retrieval”, “retrieved”, “retrieves” all get stemmed to retrieve. Many stemming programs achieve this result in somewhat of a crude approach merely by deleting the endings of words. Stemming also does not take word context into consideration and thus, stemming can result in a loss of meaning. For example, the sentence ”Programmers program with programming languages” stems to ”program program with program language.” Consideration of the dataset.

Lemmatization attempts to achieve the same thing as stemming but in a different way. Whereas stemming does not take context into account, lemmatization does. Lemmatization uses parts of speech to determine how to convert a word based on whether it is a noun, verb, pronoun, adverb, or adjective. For example the word ”better” lemmatizes to ”good” and ”corpora” to ”corpus”.

Research in [41] reports that while stemming slightly outperforms lemmatization, the differences is statistically insignificant. From manual inspection, stemming was observed to be aggressive to the datasets and we, therefore, do not perform stemming or lemmatization.

4.3 Tweet Pooling

We propose to pair LDA2Vec with temporal tweet pooling (LDA2Vec-TTP) as a means to effectively address the short length of tweet documents and we compare our approach with two other tweet pooling schemes to assess its effectiveness. We use no pooling, author-based pooling, and temporal-based pooling with a window size of one day to aggregate the tweets in our data sets.

We decided not to use hashtag-based pooling since the authors in [39] have previously created a hashtag network with this data set. Moreover, since we are investigating the underlying topics within the #BlackLivesMatter and #AllLivesMatter datasets, any hash-

tags within those tweets should support those hashtags. We do not use burst-score pooling since we are not investigating trending topic tweets. And we do not use conversation-based since we did not collect tweets and their replies and therefore do not have access to whole conversations.

4.4 Algorithm Hyperparameter Settings

We evaluate the performance of our topic modeling algorithms with their respective hyperparameters set as listed below. We also set each model to return the eight most popular topics in our datasets.

- LDA: This algorithm accepts two hyperparameters: α (topic-document density) and β (word-topic density). We determined that the settings which yield the best coherence are α = 'asymmetric' and β = 0.9.
- Biterm: We use the α and β settings used in the original paper [21] α = 50/number of topics and β = 0.01.
- LDA2Vec: This algorithm uses the β parameter for negative sampling and we set β = 0.75 as in the original paper. This setting slightly emphasizes choosing infrequent words for negative samples [38].

4.5 Evaluation

We evaluate the effectiveness of our tweet pooling schemes by comparing the coherence [42] scores, using the four coherence measures listed below, of the algorithms on each pool of tweets. Topics are considered to be coherent if all or most of the words, in a topic's top N words, are related [36].

We use the Palmetto application to calculate coherence. Palmetto is a tool which tries to help researchers by offering different coherence calculations for a topic's top words. These coherence values are based on word co-occurrences in the English Wikipedia and have been proven to correlate with human ratings. [43].

- C_p is based on a sliding window, a one-preceding segmentation of the top words and the confirmation measure of Fitelson's coherence. Word co-occurrence counts for the given top words are derived using a sliding window and the window size 70. For every top word, the confirmation to its preceding top word is calculated using the confirmation measure of Fitelson's coherence. The coherence is the arithmetic mean of the confirmation measure results [42].
- C_{UCI} is based on a sliding window and the pointwise mutual information (PMI) of all word pairs of the given top words. The word co-occurrence counts are derived using a sliding window with the size 10. For every word pair the PMI is calculated. The arithmetic mean of the PMI values is the result of this coherence [44].
- C_{UMass} is based on document co-occurrence counts, a one-preceding segmentation and a logarithmic conditional probability as confirmation measure. The main idea of this coherence is that the occurrence of every top word should be supported by every top preceding top word. Thus, the probability of a top word to occur should be higher if a document already contains a higher order top word of the same topic. Therefore, for every word the logarithm of its conditional probability is calculated using every other top word that has a higher order in the ranking of top words as condition. The probabilities are derived using document co-occurrence counts. The single conditional probabilities are summarized using the arithmetic mean [45].
- C_{NPMI} is an enhanced version of the C_{UCI} coherence using the normalized pointwise mutual information (NPMI) instead of the pointwise mutual information (PMI) [46].

5 Results

We use two datasets in this experiment containing tweets with #BlackLivesMatter and #AllLivesMatter hashtags, respectively.

5.1 #AllLivesMatter Data Statistics

Table 2 and figures 7 and 8 show #BlackLivesMatter dataset statistics, tweet length distribution and word cloud representing the 30 most popular words in the dataset.

#AllLivesMatter Dataset Statistics	
Number of tweets before pre-processing	101,576
Number of tweets after pre-processing	13,949
Unique authors	11,857
Total Words	116,071
Vocabulary Size	18,297
No Pool Avg Tweet Length	6
Author Pool Avg Tweet Length	10
Temporal Pool Avg Tweet Length	370

Table 2: #AllLivesMatter Dataset statistics

5.2 #BlackLivesMatter Data Statistics

Table 3 and figures 9 and 10 show #BlackLivesMatter dataset statistics, tweet length distribution and word cloud representing the 30 most popular words in the dataset.

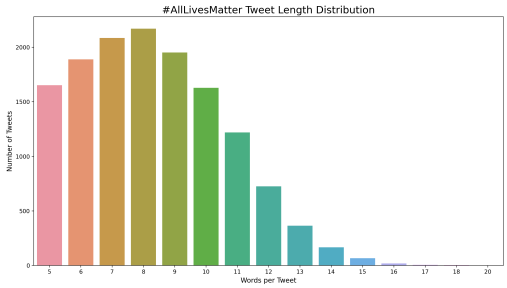

Figure 7: #AllLivesMatter Tweet Stats

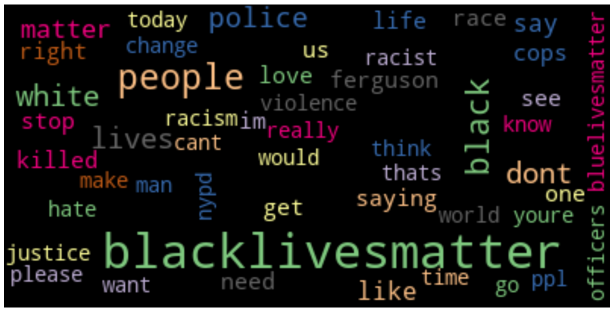

Figure 8: #AllLivesMatter word cloud

#BlackLivesMatter Dataset Statistics	
Number of tweets before preprocessing	768, 582
Number of tweets after preprocessing	127,333
Unique authors	72,062
Total Words	1,057,141
Vocabulary Size	72,941
No Pool Avg Tweet Length	8
Author Pool Avg Tweet Length	15
Temporal Pool Avg Tweet Length	2724

Table 3: #BlackLivesMatter Dataset statistics

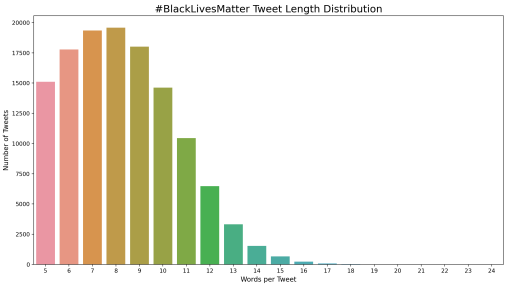

Figure 9: #BlackLivesMatter Tweet Stats

Figure 10: #BlackLivesMatter word cloud

5.3 Coherence Evaluation

Tables 4 and 5 show the coherence values produced by LDA, Biterm, and LDA2Vec, using the three pooling schemes. The bold values represent the best performing algorithm for each pooling scheme. The ”best” coherence values are the largest one relative to the others.

#AllLivesMatter Coherence

No Pool	C_p	C_{UCI}	C_{UMass}	C_{NPMI}
LDA	-0.367	-1.533	-3.023	-0.050
Biterm	**-0.252**	**-0.888**	**-1.897**	**-0.024**
LDA2Vec	-0.272	-2.161	-2.881	-0.130
Author Pool	C_p	C_{UCI}	C_{UMass}	C_{NPMI}
LDA	-0.331	-1.745	-3.223	-0.058
Biterm	**-0.231**	-1.184	**-2.053**	**-0.038**
LDA2Vec	-0.234	**-0.755**	-2.242	-0.149
Temporal Pool	C_p	C_{UCI}	C_{UMass}	C_{NPMI}
LDA	**-0.293**	**-0.752**	**-1.161**	-0.027
Biterm	**-0.293**	-0.959	-1.641	**-0.026**
LDA2Vec	-0.388	-1.155	-2.374	-0.037

Table 4: Table displaying the coherence values from the #AllLivesMatter dataset. The bold values represent the best performing algorithm for that pooling scheme

5.3.1 #AllLivesMatter Dataset Evaluation

We observe in table 4 that the Biterm topic model outperforms LDA and LDA2Vec in the dataset with no tweet pooling. The Biterm algorithm was developed specifically for topic modeling on short documents and so it follows that it performs best when tweets are not pooled and the document length is shortest.

BLM Coherences				
No Pool	**Cp**	**Cuci**	**CUmass**	**Cnpmi**
LDA	-0.206	-2.595	-3.595	-0.076
Biterm	-0.138	**-0.843**	**-2.223**	-0.018
LDA2Vec	**-0.123**	-1.111	-3.437	**-0.017**
Author Pool	**Cp**	**Cuci**	**CUmass**	**Cnpmi**
LDA	-0.180	-1.221	-2.507	-0.033
Biterm	**-0.145**	**-1.061**	**-2.319**	-0.024
LDA2Vec	-0.225	-1.131	-4.020	**-0.021**
Temporal Pool	**Cp**	**Cuci**	**CUmass**	**Cnpmi**
LDA	-0.219	-1.961	-2.737	-0.076
Biterm	**-0.210**	-1.707	-2.849	-0.058
LDA2Vec	-0.380	**-0.659**	**-1.571**	**-0.021**

Table 5: Table displaying the coherence values from the #BlackLivesMatter dataset. The bold values represent the best performing algorithm for that pooling scheme

In the author pooling scheme, Biterm performed best with all the coherence measures except for C_{UCI} where LDA2Vec had the highest score. Document lengths are still short in this pooling scheme so it follows that Biterm performed as well as it did, but surprising that LDA2Vec performed best using C_{UCI}.

In the temporal pooling scheme LDA performs best in each coherence measure except for C_{NPMI} where Biterm had the best coherence score. Biterm also matched scores with LDA for C_p. LDA2Vec performed worst in all four of the coherence measures in this pooling scheme even though the document size became much larger. We investigate this more later in the discussion section. This behavior was, however, not mirrored in the #BlackLivesMatter dataset.

5.3.2 #BlackLivesMatter Dataset Evaluation

The Biterm model performed best in the no tweet pooling dataset scoring best in C_{Umass} and C_{uci} coherence and slightly below LDA2Vec in C_{NPMI}. LDA2Vec also scored best in C_p coherence in this pool. LDA performed worst in all four coherence measures.

In the author pool, Biterm continued to perform best, as expected, with the highest coherence in all measures except C_{NPMI} where it performed slightly worse than the LDA2vec algorithm which had the best score.

In the temporal pool, LDA2Vec performed best and have the highest coherence score for each measure except C_p. This behavior is consistent with the expectation that LDA2Vec should perform better with a larger document size. Biterm had the best coherence score for the C_p measure. Surprisingly, LDA was outperformed by Biterm in all coherence measures except C_{UMass}.

5.4 Coherence Trends

Our expectation for this experiment was that we would observe coherence scores for LDA and LDA2Vec improve as document size increased, and that is what we did observe. Even though LDA2Vec performed the worst of our three algorithms in the temporal pool of the #AllLivesMatter dataset and LDA performed the worst of our three algorithms in the temporal pool of the #BlackLivesMatter dataset, both algorithm's coherence scores did improve from the no pooling coherence scores. We also observe that the coherence scores for Biterm decline as document size increases. Figures 11, 12, and 13 show the coherence trends for the #BlackLivesMatter dataset.

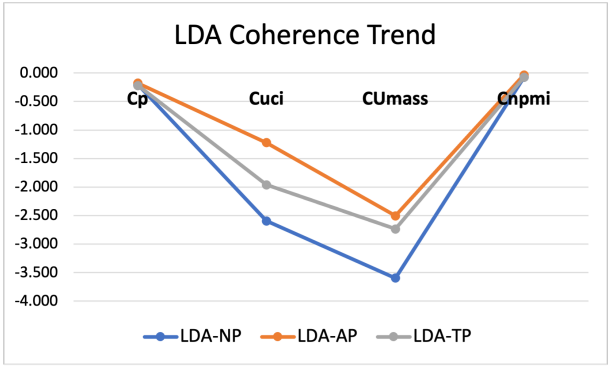

Figure 11: #BlackLivesMatter dataset LDA coherence trend curves

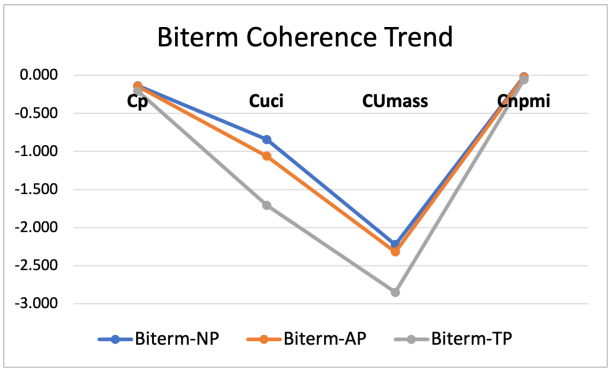

Figure 12: #BlackLivesMatter dataset Biterm coherence trend curves

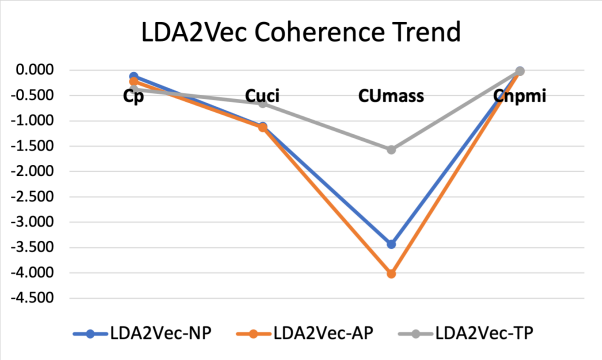

Figure 13: #BlackLivesMatter dataset LDA2Vec coherence trend curves

5.5 Topic Assignment & Human Intepretability

The effectiveness of a topic modeling algorithm is also judged based on the human interpretability of the topics it produces. We will now look at the topic assignments for the Biterm topic model and LDA2Vec-TTP from the temporal pool of the #BlackLivesMatter dataset and from the no pooling pool from the #AllLivesMatter dataset to compare the interpretability of the topics produced. Topics are considered to be coherent if all or most of the words in the word-topic assignment are related.

LDA2Vec Word-Topic Assignment #BlackLivesMatter Temporal Pooling	
Topic	Top Tokens
1	sandrabland justiceforsandrabland sayhername bland sandra whathappenedtosandrabland ripsandrabland justiceforsandy mbl samdubose
2	black police mckinney tonyrobinson icantbreathe protest mlkday selma ferguson reclaimmlk
3	ericgarner icantbreathe alllivesmatter garner black nypd eric shutitdown police protest
4	westand impct discharged whatwillbeenough infographics epartment tntj thamovement mendelson hugo
5	bernie sanders gopdebate christiantaylor berniesanders new activists hillary feelthebern blm
6	sotu whoisburningblackchurches naacpbombing stoptheparade blackchurchesburning itsmymall nigeria churches france paris
7	ferguson walterscott fergusondecision antoniomartin justiceformikebrown fergusonoctober blackoutblackfriday mikebrown wilson black
8	baltimore freddiegray baltimoreuprising charlestonshooting baltimoreriots charleston blackspring justiceforfreddiegray prayforbaltimore grammys

Figure 14: LDA2Vec topic assignment for #BlackLivesMatter temporal pooling

Biterm Word-Topic Assignment #BlackLivesMatter Temporal Pooling	
Topic	Top Tokens
1	black alllivesmatter police people ferguson white lives dont us icantbreathe
2	black alllivesmatter police ferguson people white lives dont us matter
3	cops sandrabland killed nypd today shooting murder protesters support sayhername
4	ferguson people police justice dont mikebrown get protest still sandrabland
5	eric garner alllivesmatter cant breathe man justice movement like death
6	black icantbreathe eric garner alllivesmatter people protest baltimore movement cops
7	alllivesmatter movement protest march icantbreathe eric garner ferguson shut down
8	black movement today like millions march nyc muslimlivesmatter new say

Figure 15: Biterm topic assignment for #BlackLivesMatter temporal pooling

LDA2Vec-TTP produced better coherence scores than Biterm topic model in the #BlackLivesMatter dataset with temporal pooling and we see from the word-topic assignment for both algorithms, in figures 14 and 15, that LDA2Vec-TTP produced the more interpretable topics. For example, some of the topics can be categorized as follows; topic 1: Sandra Bland; topic 3: protests for Eric Garner; topic 5: voting in the upcoming election; topic 8: protests for Freddie Gray in Baltimore. We observe a logical coherence in the topics produced. The Biterm topic model also produces interpretable topics. Some of the topics also seem to have overlap. For example, topics 1 through 6 present similar themes about police killings involving people of color (POC). Topic 8 can be categorized as violence against Muslims.

We now look at the word-topic assignment for LDA2Vec and Biterm topic model for the #AllLivesMatter dataset with no pooling.

LDA2Vec Word-Topic Assignment #AllLivesMatter No Pooling	
Topic	Top Tokens
1	syria media western movie tcot assadcrimes eerily holder obama desigualdad
2	protesta video blast york planned parenthood illuminate directly handsup pay
3	blacklivesmatter people lives say dont black saying matter like need
4	eric mikebrown babies liu rafael ramos trayvonmartin justicegarner fallen saluting
5	fatal farm bacon healthyfood realpigfarming govegan foodchat animalcruelty jessicachambers shooting
6	police killed shot cops unarmed man teen cop killedbypolice officers
7	enforcement massacred night boko haram cell genocide bulls wednesday yellow
8	world peace pray family color friends senseless human families prayers

Figure 16: LDA2Vec topic assignment for #AllLivesMatter no pooling

Biterm Word-Topic Assignment #AllLivesMatter No Pooling	
Topic	Top Tokens
1	blacklivesmatter people black lives white dont matter say like saying
2	police killed white black people blacklivesmatter officers cops shot man
3	blacklivesmatter police dont people ferguson cops black see world us
4	blacklivesmatter people one us love race life stop black peace
5	blacklivesmatter ericgarner ferguson icantbreathe police mikebrown today justice day nyc
6	blacklivesmatter people love hate event dont get one ppl us
7	blacklivesmatter prolife farm govegan defundpp babies american tcot us please
8	blacklivesmatter love see people cant ferguson like black life america

Figure 17: Biterm topic assignment for #AllLivesMatter no pooling

The Biterm topic model produced higher coherence scores than LDA2Vec for this dataset. Figures 16 and 17 show the word-topic assignment for both algorithms and we observe that there is a logical coherence to the Biterm topics. Some of the topics can be categorized as follows; topic 2: police violence; topic 3: Ferguson; topic 4: call for unity; topic 5: Eric Garner and Mike Brown; topic 7: pro-life.

The LDA2Vec algorithm did not perform well on this dataset, as expected since data sparsity will be an issue with un-pooled data. Even though we do observe some logical coherence in the word-topic assignment some of the topics are uninterpretable. For example, some of the topics can be categorized as follows; topic 1: Syrian massacre; topic 4: victims of police killings, topic 5: violence against animals; topic 6: police unarmed killings. These topics do point to issues being discussed during the time window the tweets were collected. The remainder of the topics

are relatively uninterpretable.

6 Conclusions

To our knowledge, this experiment is the first to use LDA2Vec to conduct topic modeling on tweets. The work in [28] compares Biterm Topic Model with LDA and Twitter-LDA using user-based aggregation and evaluates coherence using PMI (Pointwise Mutual Information). Our work uses four coherence measures to compare the performances of Biterm, LDA, and LDA2Vec using our tweet pooling schemes, and therefore provides a robust comparison of the respective algorithm performances.

The results of our experiments demonstrate that LDA2Vec-TTP can be used effectively for topic modeling on tweets and analyzing longstanding hashtag conversations on Twitter. Specifically, it produced logically-coherent and highly-interpretable topics with the best coherence scores in three of the four coherence measures used when analyzing the #BlackLivesMatter dataset. However, our results also showed that Biterm tends to perform better for small document sizes but that LDA and LDA2Vec tend to produce higher coherence scores as document size increases.

Future work will attempt to better characterize the variables that determine the document size at which this transition occurs.

References:

[1] S. Likhitha, B. S. Harish, and H. M. Keerthi Kumar, "A Detailed Survey on Topic Modeling for Document and Short Text Data," Tech. Rep. 39, 2019.

[2] J. Schneider, "Topic Modeling based on Keywords and Context," 10 2017.

[3] E. Jónsson and J. Stolee, "An Evaluation of Topic Modelling Techniques for Twitter," tech. rep.

[4] L. Guo, C. J. Vargo, Z. Pan, W. Ding, and P. Ishwar, "Big social data analytics in journalism and mass communication: Comparing dictionary-based text analysis and unsupervised topic modeling," *Journalism and Mass Communication Quarterly*, vol. 93, no. 2, pp. 322–359, 2016.

[5] G. Angiani, L. Ferrari, T. Fontanini, P. Fornacciari, E. Iotti, F. Magliani, and S. Manicardi, "A Comparison between Preprocessing Techniques for Sentiment Analysis in Twitter," tech. rep.

[6] A. Zubiaga, D. Spina, R. Martínez, and V. Fresno, "Real-Time Classification of Twitter Trends," tech. rep.

[7] N. Keane, C. Yee, and L. Zhou, "Using Topic Modeling and Similarity Thresholds to Detect Events," tech. rep., 2015.

[8] D. Nolasco and J. Oliveira, "Subevents detection through topic modeling in social media posts," *Future Generation Computer Systems*, vol. 93, pp. 290–303, 4 2019.

[9] V. Chakkarwar and S. Tamane, "Social Media Analytics during Pandemic for Covid19 using Topic Modeling," in *Proceedings of the 2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing, ICSIDEMPC 2020*, pp. 279–282, Institute of Electrical and Electronics Engineers Inc., 10 2020.

[10] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li, "Comparing Twitter and Traditional Media Using Topic Models," tech. rep.

[11] Monica Anderson, Skye Toor, Lee Rainie, and Aaron Smith, "An analysis of #BlackLivesMatter and other Twitter hashtags related to political or social issues," tech. rep., Pew Research Center.

[12] T. H. Chen, S. W. Thomas, and A. E. Hassan, "A survey on the use of topic models when mining software repositories," *Empirical Software Engineering*, vol. 21, pp. 1843–1919, 10 2016.

[13] M. Nadeem, M. Horn, G. Coppersmith, J. Hopkins University, and S. Sen, "Identifying Depression on Twitter," tech. rep.

[14] "Hashtag activism."

[15] P. W. Laham, "Introduction to Latent Semantic Analysis," tech. rep., 1998.

[16] T. Hofmann, "Probabilistic Latent Semantic Analysis," tech. rep.

[17] S. T. Dumais, "Latent Semantic Analysis," 2004.

[18] D. M. Blei, A. Y. Ng, and J. B. Edu, "Latent Dirichlet Allocation Michael I. Jordan," tech. rep., 2003.

[19] H. M. Wallach, "Topic Modeling: Beyond Bag-of-Words," tech. rep.

[20] M. A. Haidar and D. O'shaughnessy, "PLSA ENHANCED WITH A LONG-DISTANCE BIGRAM LANGUAGE MODEL FOR SPEECH RECOGNITION," tech. rep.

[21] X. Cheng, X. Yan, Y. Lan, and J. Guo, "IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. X, NO. X, X XXXX 1 BTM: Topic Modeling over Short Texts," tech. rep.

[22] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The Author-Topic Model for Authors and Documents," tech. rep.

[23] L. Alsumait, D. Barbará, and C. Domeniconi, "On-Line LDA: Adaptive Topic Models for Mining Text Streams with Applications to Topic Detection and Tracking," tech. rep.

[24] J. Reisinger, A. Waters, B. Silverthorn, and R. J. Mooney, "Spherical Topic Models," tech. rep., 2010.

[25] S. Lee, J. Kim, and S. H. Myaeng, "An extension of topic models for text classification: A term weighting approach," in *2015 International Conference on Big Data and Smart Computing, BIGCOMP 2015*, pp. 217–224, Institute of Electrical and Electronics Engineers Inc., 3 2015.

[26] K. Yang, Y. Cai, Z. Chen, H.-F. Leung, and R. Lau, "Exploring Topic Discriminating Power of Words in Latent Dirichlet Allocation," tech. rep.

[27] A. T. Wilson and P. A. Chew, "Term Weighting Schemes for Latent Dirichlet Allocation," tech. rep., 2010.

[28] W. Chen, J. Wang, Y. Zhang, H. Yan, and X. Li, "User Based Aggregation for Biterm Topic Model," tech. rep., 2015.

[29] V. Kumar and R. Sridhar, "Unsupervised Topic Modeling for Short Texts Using Distributed Representations of Words," tech. rep., 2015.

[30] K. Sasaki, T. Yoshikawa, and T. Furuhashi, "Online Topic Model for Twitter Considering Dynamics of User Interests and Topic Trends," tech. rep., 2014.

[31] Association for Computing Machinery. Special Interest Group on Information Retrieval., *SIGIR '13 : the proceedings of the 36th International ACM SIGIR Conference on Research & Development in Information Retrieval : July 28-August 1, 2013, Dublin, Ireland*. ACM, 2013.

[32] B. D. Davison, T. Suel, N. Craswell, B. Liu, and Association for Computing Machinery. Special Interest Group on Information Retrieval., *Proceedings of the third ACM International Conference on Web Search and Data Mining : 2010, New York, New York, USA, February 04-06, 2010*. ACM Press, 2010.

[33] D. Alvarez-Melis and M. Saveski, "Topic Modeling in Twitter: Aggregating Tweets by Conversations," tech. rep., 2016.

[34] J. Chang, J. Boyd-Graber, S. Gerrish, C. Wang, and D. M. Blei, "Reading Tea Leaves: How Humans Interpret Topic Models," tech. rep.

[35] P. Clarkson and T. Robinson, "TOWARDS IMPROVED LANGUAGE MODEL EVALUATION MEASURES," tech. rep.

[36] S. Syed and M. Spruit, "Full-Text or abstract? Examining topic coherence scores using latent dirichlet allocation," in *Proceedings - 2017 International Conference on Data Science and Advanced Analytics, DSAA 2017*, vol. 2018-January, pp. 165–174, Institute of Electrical and Electronics Engineers Inc., 7 2017.

[37] K. Stevens, P. Kegelmeyer, D. Andrzejewski, and D. Buttler, "Exploring Topic Coherence over many models and many topics," tech. rep., 2012.

[38] C. E. Moody, "Mixing Dirichlet Topic Models and Word Embeddings to Make lda2vec," 5 2016.

[39] R. J. Gallagher, A. J. Reagan, C. M. Danforth, and P. S. Dodds, "Divergent discourse between protests and counter-protests: #BlackLivesMatter and #AllLivesMatter," *PLoS ONE*, vol. 13, 4 2018.

[40] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," *Information Processing and Management*, vol. 50, no. 1, pp. 104–112, 2014.

[41] K. Kettunen, T. Kunttu, and K. Järvelin, "To stem or lemmatize a highly inflectional language in a probabilistic IR environment?," *Journal of Documentation*, vol. 61, no. 4, pp. 476–496, 2005.

[42] M. Röder, A. Both, and A. Hinneburg, "Exploring the space of topic coherence measures," in *WSDM 2015 - Proceedings of the 8th ACM International Conference on Web Search and Data Mining*, pp. 399–408, Association for Computing Machinery, Inc, 2 2015.

[43] Michael Röder, "Palmetto is a quality measuring tool for topics," 2016.

[44] D. Newman, □. □. Jey, H. Lau, K. Grieser, and T. Baldwin, "Automatic Evaluation of Topic Coherence," tech. rep., 2010.

[45] D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. Mccallum, "Optimizing Semantic Coherence in Topic Models," tech. rep., 2011.

[46] N. Aletras and M. Stevenson, "Evaluating Topic Coherence Using Distributional Semantics," tech. rep.

Follow: www.wseas.org/multimedia/contributor-role-instruction.pdf

Creative Commons Attribution License 4.0 (Attribution 4.0 International , CC BY 4.0)

Contribution of individual authors to the creation of a scientific article (ghostwriting policy)

Kristofferson Culmer collected the data and carried out the analysis.
Dr. Jeffrey Uhlmann advised on the process of formulating the problem, carrying out the experiment, and analyzing the data.