ATM: Adversarial-neural Topic Model

Rui Wang
Southeast University
Nanjing, China, 210096
rui_wang@seu.edu.cn

Deyu Zhou*
Southeast University
Nanjing, China, 210096
d.zhou@seu.edu.cn

Yulan He
University of Warwick
Coventry CV4 7AL, UK
yulan.he@warwick.ac.uk

Abstract

Topic models are widely used for thematic structure discovery in text. But traditional topic models often require dedicated inference procedures for specific tasks at hand. Also, they are not designed to generate word-level semantic representations. To address these limitations, we propose a topic modeling approach based on Generative Adversarial Nets (GANs), called Adversarial-neural Topic Model (ATM). The proposed ATM models topics with Dirichlet prior and employs a generator network to capture the semantic patterns among latent topics. Meanwhile, the generator could also produce word-level semantic representations. To illustrate the feasibility of porting ATM to tasks other than topic modeling, we apply ATM for open domain event extraction. Our experimental results on the two public corpora show that ATM generates more coherence topics, outperforming a number of competitive baselines. Moreover, ATM is able to extract meaningful events from news articles.

1 Introduction

Topic models Blei (2012) underpin many successful applications within the field of Natural Language Processing (NLP). Variants of topic models have been proposed for different tasks including topic-associated sentiment analysis Lin and He (2009) and event extraction from social media Zhou, Chen, and He (2014). However, topic models typically rely on mean-field variational inference or collapsed Gibbs sampling for model learning. A small change to the modeling assumption requires the re-derivation of the whole inference algorithm, which is mathematically arduous and time consuming.

*the superscript means the corresponding author.

In recent years, word embeddings have gained an increasing interest thanks to their improved efficiency in representing words as continuous vectors in a low-dimensional space Mikolov et al. (2013); Joulin et al. (2016); Bojanowski et al. (2016); Athiwaratkun, Wilson, and Anandkumar (2018). The resulting embeddings encode numerous semantic relations (similarity or analogies). But the traditional topic models could not generate such word-level semantic representations.

To overcome these limitations, Neural Variational Document Model (NVDM) Miao, Yu, and Blunsom (2016) used a hidden layer to reconstruct the document by generating the words independently and employed a semantic matrix to represent the word embeddings. However, the use of Gaussian prior over topics may lead to incoherent topics being generated. On the contrary, Srivastava and Sutton (2017) proposed LDA-VAE, a neural topic model based on the variational autoencoder (VAE), in which the logistic normal distribution was used as the prior for topic generation. But the proposed model is not able to produce word-level semantic representations.

In this paper, we propose the Adversarial-neural Topic Model (ATM) based on adversarial training. The principle idea is to use a generator network to learn the projection function between the document-topic distribution and the document-word distribution. Instead of providing an analytic approximation, as in traditional topic models, the ATM uses a discriminator network to recognize if the input document is real or fake and helps the generator to construct a more realistic document from a random noise drawn from a Dirichlet distribution. Due to the flexibility of neural networks, the generator is capable of learning complicated non-linear distributions. And the supervision provided by the discriminator in the adversarial training phase will help generator to capture the semantic patterns embedded in the latent topics. Besides, the connection weights between the embedding layer and the word distribution layer of the generator also encodes the semantic information and naturally provides distributed representations of words.

Our contributions are summarized below:

- We propose a novel Adversarial-neural Topic Model (ATM), which is, to the best of our knowledge, the first attempt of using adversarial training for topic modeling.
- Unlike traditional topic models, ATM is able to not only extract topics from text, but also naturally produce word embeddings encoding the semantic relations among words.
- Experimental results on two public datasets show that ATM outperforms the state-of-the-art approaches in terms of topic coherence measure. In addition, the result on the open domain event extraction dataset verifies the portability of ATM.

2 Related Work

Our work is related to two lines of research, neural-based topic modeling and the Generative Adversarial Nets.

Neural-based Topic Modeling

To overcome the difficult exact inference of topic models based on directed graph, Hinton and Salakhutdinov (2009) modified the Restricted Boltzmann Machines and proposed a replicated softmax model (called RSM). Inspired by the variational autoencoder, Miao, Yu, and Blunsom (2016) used the multivariate Gaussian as the prior distribution of latent space and proposed the Neural Variational Document Model (NVDM) for text modeling. More recently, to deal with the inappropriate Gaussian prior of NVDM, Srivastava and Sutton (2017) proposed LDA-VAE which approximated the Dirichlet prior using a logistic normal distribution. They also presented ProdLDA which further improved topic coherence.

Generative Adversarial Nets

As a neural-based generative model, the Generative Adversarial Nets Goodfellow et al. (2014) have been extensively researched from both theoretical and practical aspects.

Theoretically, Nowozin, Cseke, and Tomioka (2016) used the Fenchel conjugate to define the F-divergence and proposed the F-GAN to generalize its optimization objective. To precisely measure the distance between two high dimensional distributions, Arjovsky, Chintala, and Bottou (2017) defined the Earth Mover's Distance (Wasserstein distance) and gave a computational method based on the weight clipping mechanism. Along this line, Gulrajani et al. (2017) improved the Wasserstein GAN by adding a gradient penalty loss and promoted the stability of adversarial training.

In practical applications, many variants of GAN have been developed for NLP tasks. Such as text generation, a hot research area in NLP. The sequence generative adversarial network (SeqGAN) proposed in Yu et al. (2017) incorporated a policy gradient strategy to optimize the generation process. Based on the policy gradient, Lin et al. (2017) proposed the RankGAN to capture the rich structures of language by ranking and analysing a collection of human-written and machine-written sentences. To overcome the mode collapse when dealing with discrete data, Fedus, Goodfellow, and Dai (2018) proposed the MaskGAN which used an actor-critic conditional GAN to fill in missing text conditioned on the surrounding context. Along this line, Wang and Wan (2018) proposed the SentiGAN to generate texts of different sentiment labels. Besides, Miyato, Dai, and Goodfellow (2016); Li and Ye (2018) improved the performance of semi-supervised text classification using adversarial training, Zeng et al. (2018); Qin, Xu, and Wang (2018) designed GAN-based models for distance supervision relation extraction.

Despite many successful applications using GAN-based approaches, none of these approaches tackles the topic modeling problem. We propose the first GAN-based topic model called ATM, which differs from the existing approaches to neural topic modeling in the following aspects: (1) Unlike NVDM and LDA-VAE which uses either Gaussian prior or logistic-normal prior for latent topics, ATM uses the Dirichlet prior instead; (2) A generator network is used to learn the projection function between the document-topic distribution and the document-word distribution, which essentially captures the semantic patterns among latent topics rather than generating text sequences; (3) ATM is able to generate word-level semantic representations as a side product.

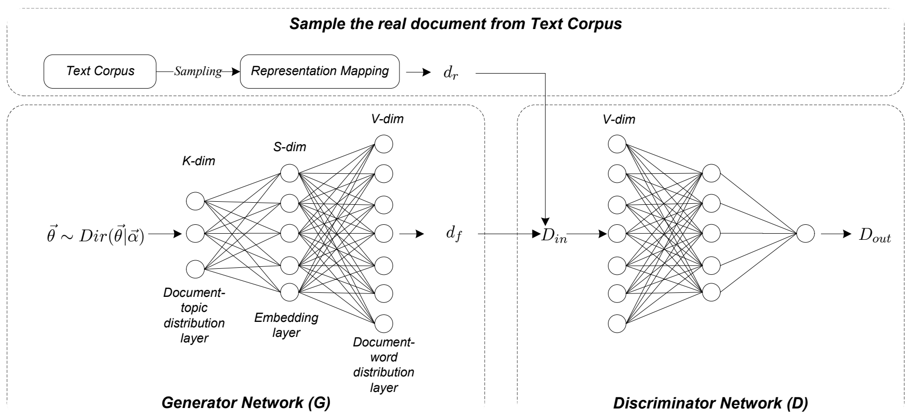

Figure 1: The framework of the Adverarial-neural Topic Model (ATM).

3 Adversarial-neural Topic Model

We propose the Advesarial-neural Topic Model (ATM) as shown in Figure 1. The proposed ATM contains three main components: (1) the document sampling module shown at the top of Figure 1, which defines the representation mapping function and samples a real document $d_r \in \mathbb{R}^V$ from an input text corpus; (2) the generator G takes a topic distribution $\vec{\theta}$ sampled from a Dirichlet prior as input and generates the corresponding fake document d_f; (3) the discriminator D takes d_f and d_r as input and discriminates the fake document from the real ones, whose output is subsequently used as a learning signal to update the parameters of G and D. We explain the design and function of each of these modules in more details below.

3.1 Representation Mapping

Each document d is represented by a normalized V-dimensional vector weighted by TF-IDF. More concretely:

$$tf_{i,d} = \frac{n_{i,d}}{\sum_v n_{v,d}} \qquad idf_i = \log \frac{|C|}{|C_i|}$$

$$tf\text{-}idf_{i,d} = tf_{i,d} \times idf_i \qquad d_r^i = \frac{tf\text{-}idf_{i,d}}{\sum_v tf\text{-}idf_{v,d}}$$

where V is the vocabulary size, $n_{i,d}$ denotes the number of times the i-th word appears in document d, $|C|$ denotes the total number of documents in the corpus, and $|C_i|$ is the number of documents containing the i-th word. With this representation, each document in the corpus could be regarded as a multinomial distribution over V words, and each dimension reflects the semantic coherence between the i-th word and the document d.

3.2 Network Architecture

The G network contains three layers, the K-dimensional document-topic distribution layer, the S-dimensional embedding layer and the V-dimensional document-word distribution layer as shown in Figure 1. First, the G network takes a randomly sampled topic distribution $\vec{\theta}$ as input and transforms it into a document-word distribution. To model the multinomial property of the document-topic distribution, $\vec{\theta}$ is drawn from $Dir(\vec{\theta}|\vec{\alpha})$:

$$p(\vec{\theta}|\vec{\alpha}) = Dir(\vec{\theta}|\vec{\alpha}) \triangleq \frac{1}{\triangle(\vec{\alpha})} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1} \tag{1}$$

where $\vec{\alpha}$ is the hyper-parameter of the Dirichlet distribution, $\triangle(\vec{\alpha}) = \frac{\prod_{k=1}^{K} \Gamma(\alpha_k)}{\Gamma(\sum_{k=1}^{K} \alpha_k)}$, K is the number of topics, $\theta_k \in [0,1]$ denotes the proportion of topic k in the document and $\sum_{k=1}^{K} \theta_k = 1$.

Then, G projects $\vec{\theta}$ into the S-dimensional (set to 100 in experiments) semantic space through the embedding layer based on equations :

$$\vec{a}_s = \max((W_s\vec{\theta} + \vec{b}_s), leak * (W_s\vec{\theta} + \vec{b}_s)) \tag{2}$$

$$\vec{o}_s = BN(\vec{a}_s) \tag{3}$$

where $W_s \in \mathbb{R}^{S \times K}$ is the weight matrix and $\vec{b}_s$ represents the bias term of the embedding layer, $\vec{a}_s$ is the state vector activated by the LeakyReLU function parameterized with $leak$, BN denotes batch normalization and $\vec{o}_s$ is the output of the embedding layer.

Finally, G transforms $\vec{o}_s$ to a V-dimensional multinomial distribution d_f using :

$$\vec{h}_w = W_w\vec{o}_s + \vec{b}_w \tag{4}$$

$$o_w^i = \frac{\exp(h_w^i)}{\sum_{v=1}^{V} \exp(h_w^v)} \tag{5}$$

where $W_w \in \mathbb{R}^{V \times S}$ learns the semantic word embeddings and $\vec{b}_w$ represents the bias term, $\vec{h}_w$ is the state vector and o_w^i denotes the probability of i-th word in d_f.

Likewise, we design the discriminator as a three layer fully connected network. The D network employs the d_f and the d_r as input and outputs a scalar as shown in Figure 1. A higher D_{out} means that the discriminator is prone to consider the input data as a real document and vice versa.

3.3 Training

The fake document d_f and the real document d_r shown in Figure 1 could be viewed as the random sample from two V-dimensional dirichlet distribution $\mathbb{P}_g$ and $\mathbb{P}_r$. And the training objective of ATM is to let the generated distribution $\mathbb{P}_g$ approximate the real data distribution $\mathbb{P}_r$ as much as possible. Thus, the choice of divergence that measures the distance between two distributions is crucial for effective training of ATM.

The original GAN Goodfellow et al. (2014) used the Jensen-Shannon divergence as the optimization objective. However, Arjovsky, Chintala, and Bottou (2017) argued that the divergences which GANs typically minimize are potentially not continuous with respect to the generator's parameters, leading to mode collapse and training difficulty. They proposed instead using the *Earth-Mover* distance (also called Wasserstein-1) which is defined as the minimum cost of transporting mass in order to transform the distribution $\mathbb{P}_g$ into the distribution $\mathbb{P}_r$. Further, Gulrajani et al. (2017) improved the Wassertein-1 with a gradient penalty strategy which performed more stable. We follow their work and define the objective of ATM as:

$$L_d = \mathop{\mathbb{E}}_{d_f \sim \mathbb{P}_g}[D(d_f)] - \mathop{\mathbb{E}}_{d_r \sim \mathbb{P}_r}[D(d_r)] \tag{6}$$

$$L_{gp} = \mathop{\mathbb{E}}_{\hat{d} \sim \mathbb{P}_{\hat{d}}}[(\| \nabla_{\hat{d}} D(\hat{d}) \|_2 - 1)^2] \tag{7}$$

$$L = L_d + \lambda L_{gp} \tag{8}$$

where L_d and L_{gp} denote the loss of discriminator D and the gradient penalty, respectively, λ is the gradient penalty coefficient, $\hat{d}$ could be obtained by sampling uniformly along a straight line between a real document d_r and a generated document d_f, and $\mathbb{P}_{\hat{d}}$ is the distribution from which $\hat{d}$ is sampled.

Algorithm 1 Training procedure for ATM

Input: $K, \lambda, n_d, m, \alpha_1, \beta_1, \beta_2$
Output: the trained generator network G.

Initial D parameters ω_d and G parameter ω_g
while ω_g has not converged **do**
  for $t = 1, ..., n_d$ **do**
    for $j = 1, ..., m$ **do**
      Sample $d_r \sim \mathbb{P}_r$,
      Sample a random $\vec{\theta} \sim Dir(\vec{\theta}|\vec{\alpha})$
      Sample a random number $\epsilon \sim U[0, 1]$
      $d_f \leftarrow G(\vec{\theta})$
      $\hat{d} \leftarrow \epsilon d_r + (1 - \epsilon) d_f$
      $L_d^{(j)} = D(d_g) - D(d_r)$
      $L_{gp}^{(j)} = (\| \nabla_{\hat{d}} D(\hat{d}) \| - 1)^2$
      $L^{(j)} \leftarrow L_d^{(j)} + \lambda L_{gp}^{(j)}$
    end for
    $\omega_d \leftarrow Adam(\nabla_{\omega_d} \frac{1}{m} \sum_{j=1}^{m} L^{(j)}, \omega_d, p_a)$
  end for
  Sample m noise $\left\{ \vec{\theta}^{(j)} \sim Dir(\vec{\theta}|\vec{\alpha}) \right\}$
  $\omega_g \leftarrow Adam(\nabla_{\omega_g} \frac{-1}{m} \sum_{j=1}^{m} D(G(\vec{\theta}^{(j)})), \omega_g, p_a)$
end while

Based on the model structure and the optimization objective described above, the training procedure for ATM is given in Algorithm 1. Here, n_d denotes the number of discriminator iterations per generator iteration, m represents the batch size, α_1 is the learning rate, β_1 and β_2 are other hyper-parameters of Adam optimizer Kingma and Ba (2014), and p_a denotes $\{\alpha_1, \beta_1, \beta_2\}$. We use the default values of $\lambda = 10$, $n_d = 5$, $m = 512$. Moreover, the α_1, β_1 and β_2 are set to 0.0001, 0 and 0.9 respectively.

3.4 Topic Generation

The trained generator G learns the projection function between the document-topic distribution and the document-word distribution. That is, given a topic distribution $\vec{\theta_d}$ for a document d, G is able to generate the corresponding word distribution.

To generate the word distribution of each topic, we use $\vec{ts}_{(k)}$, a K-dimensional vector, as the one-hot encoding of the k-th topic. For example, $\vec{ts}_{(1)} = [1, 0, 0, 0, 0]^\intercal$ in the five topic number setting. We could then obtain the word distribution $\vec{\phi}_k$ for topic k using:

$$\vec{\phi}_k = G(\vec{ts}_{(k)}) \tag{9}$$

4 Experiments

We evaluate our proposed ATM on two tasks, topic extraction and open domain event extraction. We first describe the datasets and the baseline approaches, and then present the topic coherence evaluation results for the topic extraction task. Finally, we discuss the results of using ATM for open domain event extraction to validate the feasibility of applying ATM for tasks other than topic modeling.

4.1 Experimental Setup

Two publicly accessible datasets, Grolier[1] and NYtimes[2] datasets, are used for topic coherence evaluation, and an event dataset built based on the Global Database of Events, Language, and Tone (GDELT)[3] is used for event extraction. Details are summarized below:

- *Grolier dataset*[1] is built from Grolier Multimedia Encyclopedia, and its content covers almost all the fields in the world.
- *NYtimes dataset*[2] is a collection of news articles published between 1987 and 2007, and the dataset has a wide range of topics, such as sports, politics, economy, etc.
- *Event dataset*. We crawl and parse the GDELT Event Database[3] containing articles published on the first day of May in 2014.

We choose the following five models as the baselines:

[1] https://cs.nyu.edu/∼roweis/data/
[2] http://archive.ics.uci.edu/ml/datasets/Bag+of+Words
[3] http://data.gdeltproject.org/events/index.html

Dataset	#Document	#Words
Grolier	29,762	15,276
NYtimes	99,992	12,604
Event	20,199	9,346

Table 1: The statistics of datasets.

Dataset	Model	C_P	C_A	NPMI	UCI	UMass
Grolier	NVDM	-0.187746	0.145684	-0.061911	-2.114927	-4.291624
	LDA-VAE	-0.220548	0.150469	-0.065378	-2.479750	-4.755522
	ProdLDA	-0.037436	0.173391	-0.019347	-1.639878	-4.542689
	LDA	0.190845	0.200942	0.049753	-0.050336	-2.918612
	ATM	**0.210448**	**0.218898**	**0.058167**	**0.105086**	**-2.765081**
NYtimes	NVDM	-0.413086	0.134154	-0.143711	-4.307269	-5.931614
	LDA-VAE	-0.157560	0.148221	-0.061418	-2.420816	-4.640276
	ProdLDA	-0.003455	0.196395	-0.028223	-1.917367	-4.193377
	LDA	0.308336	0.212750	0.077278	0.516503	-2.420221
	ATM	**0.356771**	**0.237524**	**0.089874**	**0.658218**	**-2.324093**

Table 2: Average topic coherence on Grolier and NYtimes corpus with five topic settings [20, 30, 50, 75, 100].

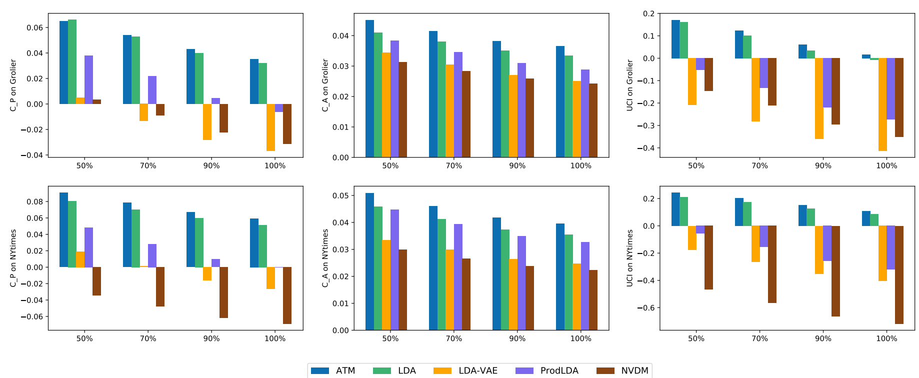

Figure 2: Average topic coherence on Grolier and NYtimes with five topic settings [20, 30, 50, 75, 100] among topics whose coherence values are ranked at the top 50%, 70%, 90% and 100% positions.

- **LDA** Blei, Ng, and Jordan (2003), is a topic model that generates topics based on word co-occurrence patterns from documents. We implement the LDA model and set the Dirichlet prior of the document-topic distribution $\alpha = 50/K$ and the Dirichlet prior of the topic-word distributions $\beta = 0.01$, following what have been suggested in Griffiths and Steyvers (2004).
- **NVDM** Miao, Yu, and Blunsom (2016), is an unsupervised text modeling ap-

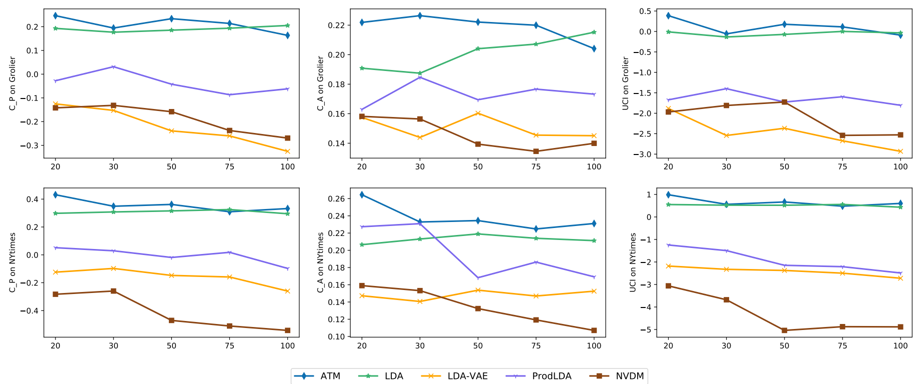

Figure 3: Average topic coherence (100%) on Grolier and NYtimes datasets vs. different topic setting [20, 30, 50, 75, 100].

Model	Topics
ATM	jet flight airline hour plane passenger trip plan travel pilot
	stock market companies money investor technology fund investment company business
	music song musical album jazz band record recording mp3 composer
	voter vote poll republican race primary percent election campaign democratic
	film movie actor director award movies character theater production play
LDA	flight plane *ship crew* air pilot hour *boat* passenger airport
	stock market *percent* investor analyst *quarter* investment shares share fund
	music song band sound record artist album show musical rock
	voter vote poll election campaign primary candidates republican race party
	film movie character play actor director movies *minutes* theater *cast*
ProdLDA	*wireless* customer *telecommunication* airlines *broadband satellites phones subscriber* airline *provider*
	brokerage securities broker lender buyer transaction investor investment stock borrower
	musical album *playwright* composer *choreographer onstage* songwriter song guitarist repertory
	voter vote votes election electoral polling poll presidential primaries *turnout*
	film comedy *beginitalic enditalic sci* filmmaker cinematic filmmaking movie starring
LDA-VAE	passenger destination traveler *fares booking* airlines luggage routes *rider excursion*
	acquisition shareholder merge *takeover acquire* merger *consolidated* stockholder *suitor consolidation*
	soloist operatic composer *repertory* troupe *choreographer choreography* sung *dances* recital
	balloting nominating election elect incumbent victor primaries *contested* electoral vote
	moviegoer studios filmmaker movies film filming *vh1* studio stardom *rapper*
NVDM	*nesting instructor ranchers wingspan veteran* fly *manager* pilot *ecosystems* flight
	company billion companies *production equipment processed processing producer manufacturing products*
	conducting conductor instrumental *interval staff* discography *knighted radioactive charge* director
	degrees national party *billion* nations *decrease university exceed disorder nuclear*
	bay film *indian french company* novel *dec lake explorer travels*

Table 3: Topic examples of all the models, italics means out-of-topic.

proach based on variational autoencoder. We use the original implementation[4].

- **LDA-VAE** Srivastava and Sutton (2017), is a neural topic model based on variational autoencoder. We use the implementation in the paper[5].
- **ProdLDA** Srivastava and Sutton (2017), is a variant of LDA-VAE, in which the distribution over individual words is a product of experts rather than the mixture

[4]https://github.com/ysmiao/nvdm

[5]https://github.com/akashgit/autoencoding_vi_for_topic_models

model used in LDA. The original implementation is used.

- **LEM** Zhou, Chen, and He (2014), is a Bayesian modeling approach for open domain event extraction. It treats an event as a latent variable and models the generation of an event as a joint distribution of its individual event elements (org, loc, per, key)[6]. We implement the algorithm with the default configuration.

For the NYtimes dataset, we random select 100,000 articles and remove the low frequent words. For the Event dataset, we use the Stanford Named Entity Recognizer[7] Finkel, Grenager, and Manning (2005) for identifying the named entities (Location, Organization and Person). In addition, we remove common stopwords and only keep the recognized name entities and the tokens which are verbs, nouns, or adjectives from these event documents. The statistics of the processed corpora are shown in Table 1.

4.2 Topic Coherence Evaluation

Typically topic models are evaluated based on the likelihood of held-out documents. However, as pointed out in Chang et al. (2009), higher likelihood of held-out document does not necessarily correspond to human judgement of topic coherence. In this subsection, we follow Röder, Both, and Hinneburg (2015) and choose five coherence metrics to evaluate the topics generated by models. They are C_P (a metric based on a sliding window, a one-preceding segmentation of the given words and the confirmation measure of Fitelson's coherence), C_A (a metric based on a context window, a pairwise comparison of the given words and an indirect confirmation measure that uses normalized pointwise mutual information and the cosine similarity), UCI (a metric based on a sliding window and the pointwise mutual information of all word pairs of the given topics), NPMI (an enhanced version of UCI using the normalized pointwise mutual information) and UMass Mimno et al. (2011) (a metric based on document cooccurrence counts, a one-preceding segmentation and a logarithmic conditional probability as confirmation measure). For all these five metrics, higher value implies more coherent topic. In our evaluation, we choose the top 10 words to represent each topic and compute the topic coherence using the Palmetto library[8].

To compare the performance of the proposed approach, experiments are conducted on Grolier and NYtimes with five topic number settings [20, 30, 50, 75, 100]. The average coherence values are listed in Table 2 and each value is computed by averaging the average topic coherences (all the topics are used) over five topic number settings. Besides, we calculate the average topic coherence among topics whose coherence values are ranked at the top 50%, 70%, 90%, 100% positions. For example, to calculate the average UCI coherence of ATM @ 70%, we first compute the average UCI coherence with the select topics whose UCI values are ranked at the top 70% positions for each topic number setting, and then average the five averaged coherence values. The corresponding results are shown in Figure 2. It can be observed from Figure 2 that the proposed model outperforms the LDA, NVDM, LDA-VAE and ProdLDA in general.

[6]means organization, location, person and keywords.
[7]https://nlp.stanford.edu/software/CRF-NER.html
[8]https://github.com/dice-group/Palmetto

To explore how topic coherence results vary with different topic numbers, we show in Figure 3 the average topic coherence of two datasets vs. different topic number settings. We can observe that ATM achieves better results compared to other baselines most of the time with 20, 30, 50 or 75 topics. However, when the topic number is 100, the performance gap between ATM and LDA diminishes and in some cases (e.g., C_P and C_A for the Grolier dataset), ATM gives slightly worse results compared to LDA, though it still largely outperforms all the other baselines. This might attribute to the increased network complexity due to the larger topic number setting.

Events	ATM	LEM
MH370	org: **air airlines** ministry transport international loc: **malaysia** beijing france vietnam dubai per: **hishammuddin hussein** najib kerry lee key: **search flight aircraft air plane**	org: **airlines air** international transport government loc: **malaysia** south korea beijing us per: **hussein hishammuddin** fitch long park key: **flight airlines plane** preliminary **search**
Saudi MERS	org: **community** ministry **saudi healthcare** government loc: **saudi** ontario iran canada jeddah per: **president** obama jordan kerry walker key: **health hospital patients disease medical**	org: **saudi** jordan army eastern state loc: east **saudi** jordan egypt israel per: jordan **president** frank rob geldof key: east middle **respiratory syndrome health**
Pakistan vs. India	org: **army kashmir** sharif taliban afghanistan loc: **pakistan kashmir india** afghanistan islamabad per: **sharif** kerry khan president lovell key: **army peace chief region** province	org: **army kashmir** sharif government congress loc: **pakistan kashmir** islamabad **india** delhi per: **sharif** tsvangirai morgan dube biti key: **army chief** vein news **peace**
Indian Election	org: **bjp party congress singh gandhi** loc: **gujarat india varanasi delhi** seemandhra per: **modi singh** gandhi naidu khan key: **congress election candidate minister leader**	org: **bjp congress party commission delhi** loc: **delhi gujarat** modis **varanasi india** per: **modi** gandhi **singh** modis president key: **prime candidate election ministerial congress**
Taksim Clash	org: **police** city **government** erdogan **union** loc: **taksim istanbul** city **turkey union** per: **erdogan** park walker quinn hall key: **square protesters tear demonstrators street**	org: **police** international **labor** central greenpeace loc: **istanbul taksim turkey** rotterdam **union** per: mark **erdogan** geldof park hall key: **protesters square** international **gas water**

Table 4: The event examples extracted by ATM and LEM.

From the above topic coherence evaluation results, it is clear that ATM is able to extract more coherence topics compared to baselines. To verify this qualitatively, we show examples of topics from all the models in Table 3. These topics correspond to *'airline'*, *'trade'*, *'music'*, *'election'* and *'film'* respectively. Words that do not seem to belong to its corresponding topic are highlighted in italic. It can be observed that the number of less semantically relevant words somewhat correlates with the coherence results observed earlier in Table 2 and Figure 2.

Unlike traditional topic models, the proposed ATM could learn the semantic embeddings of words apart from generating coherent topics. The weights matrix $W_w \in \mathbb{R}^{V \times S}$ contains the word-level semantic information, and each row could be viewed as the corresponding word embedding. Thus, we select the topic words of six topics from a 50-topic run on the NYtimes corpus and use the Principal Component Analysis (PCA) to project their word embeddings into a two-dimensional space. The visualization of these topic words is shown as Figure 4. We can clearly see that the words related to the *'trade'* topic are grouped at the lower right corner, and the topic words of *'religious'* are displayed at the top region. Besides, the words related to the topics *'music'* and *'film'* are close to each other, which is not surprising, since these topics are closely related.

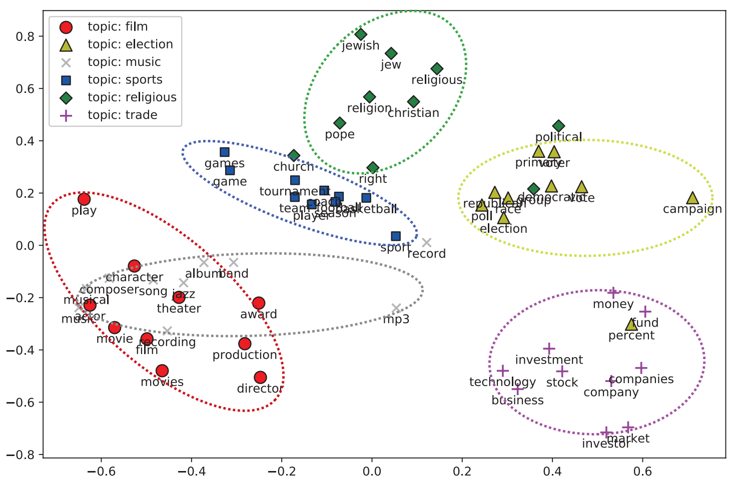

Figure 4: visualization of the topic words from the six selected topics.

4.3 Open Domain Event Extraction

To further prove the feasibility of porting ATM to tasks other than topic modeling, we apply it for open domain event extraction. For this task, an event is represented in a structured form as $< org, loc, per, key >$[6] Zhou, Xu, and He (2015), with each of the elements in the quadruples represented by a list of words.

We use the pre-identified named entities[7], verbs, nouns and adjectives to construct the word set of organization, location, person and keywords. When using ATM for event extraction, these four word sets and the event-specific word distribution are used to generate the related topics. For example, the organization topic of an event could be obtained by sorting the words in the organization word set based on the corresponding probabilities in the event-specific word distribution learned by ATM. Table 4 shows the example events extracted by ATM and LEM where the relevant words are highlighted in bold. It can be observed that ATM performs comparably with LEM. However, while LEM required the model-specific inference algorithm to be derived, ATM did not need any modification of its network architecture or parameter estimation procedure.

To validate the correctness of the extracted events, we retrieve the title of articles using the event-related words from ATM and obtain the following results:

- *Missing Malaysia Airlines flight MH370: Government report suggests official search for plane did not begin until four hours after disappearance.*
- *Saudi Arabia finds 26 more cases of MERS, Egypt reports first sufferer.*
- *India's defence experts and politicos condemn Pak Army Chief's Kashmir statement.*
- *Top BJP leaders, Rajnath Singh, MM Joshi, Sushma Swaraj to campaign for Narendra Modi in Varanasi*

- *Turkey May Day protests hit by tear gas near Taksim Square - Panorama.*

It is clear that the retrieved titles indeed correspond well with the extracted events by ATM.

5 Conclusions

We have proposed a novel topic modeling approach based on adversarial training. The proposed approach, ATM, models the topics with Dirichlet prior and employs the generator network to learn the semantic patterns among latent topics. Apart from automatically generating latent topics from a text corpus, it could also produce word-level semantic representations as a side product. The experimental comparison with the state-of-the-art methods show that ATM achieves improved topical coherence results. Moreover, the feasibility of porting ATM for tasks other than topic modeling has been verified for open domain event extraction. In the future, we plan to extend the ATM to cope with the data sparsity in short text. And another direction we are interested in exploring is to develop dynamic and correlated topic models based on adversarial training.

References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875*.

Athiwaratkun, B.; Wilson, A. G.; and Anandkumar, A. 2018. Probabilistic fasttext for multi-sense word embeddings. *arXiv preprint arXiv:1806.02901*.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3(Jan):993–1022.

Blei, D. M. 2012. Probabilistic topic models. *Communications of the ACM* 55(4):77–84.

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Chang, J.; Gerrish, S.; Wang, C.; Boyd-Graber, J. L.; and Blei, D. M. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*, 288–296.

Fedus, W.; Goodfellow, I.; and Dai, A. M. 2018. Maskgan: Better text generation via filling in the _. *arXiv preprint arXiv:1801.07736*.

Finkel, J. R.; Grenager, T.; and Manning, C. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, 363–370. Association for Computational Linguistics.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Griffiths, T. L., and Steyvers, M. 2004. Finding scientific topics. *Proceedings of the National academy of Sciences* 101(suppl 1):5228–5235.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, 5769–5779.

Hinton, G. E., and Salakhutdinov, R. R. 2009. Replicated softmax: an undirected topic model. In *Advances in neural information processing systems*, 1607–1614.

Joulin, A.; Grave, E.; Bojanowski, P.; and Mikolov, T. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, Y., and Ye, J. 2018. Learning adversarial networks for semi-supervised text classification via policy gradient. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1715–1723. ACM.

Lin, C., and He, Y. 2009. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, 375–384. ACM.

Lin, K.; Li, D.; He, X.; Zhang, Z.; and Sun, M.-T. 2017. Adversarial ranking for language generation. In *Advances in Neural Information Processing Systems*, 3155–3165.

Miao, Y.; Yu, L.; and Blunsom, P. 2016. Neural variational inference for text processing. In *International Conference on Machine Learning*, 1727–1736.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mimno, D.; Wallach, H. M.; Talley, E.; Leenders, M.; and McCallum, A. 2011. Optimizing semantic coherence in topic models. In *Proceedings of the conference on empirical methods in natural language processing*, 262–272. Association for Computational Linguistics.

Miyato, T.; Dai, A. M.; and Goodfellow, I. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.

Nowozin, S.; Cseke, B.; and Tomioka, R. 2016. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, 271–279.

Qin, P.; Xu, W.; and Wang, W. Y. 2018. Dsgan: Generative adversarial training for distant supervision relation extraction. *arXiv preprint arXiv:1805.09929*.

Röder, M.; Both, A.; and Hinneburg, A. 2015. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, 399–408. ACM.

Srivastava, A., and Sutton, C. 2017. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*.

Wang, K., and Wan, X. 2018. Sentigan: Generating sentimental texts via mixture adversarial networks. In *IJCAI*, 4446–4452.

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, 2852–2858.

Zeng, D.; Dai, Y.; Li, F.; Sherratt, R. S.; and Wang, J. 2018. Adversarial learning for distant supervised relation extraction. *Computers, Materials & Continua* 55(1):121–136.

Zhou, D.; Chen, L.; and He, Y. 2014. A simple bayesian modelling approach to event extraction from twitter. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, 700–705.

Zhou, D.; Xu, H.; and He, Y. 2015. An unsupervised bayesian modelling approach for storyline detection on news articles. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1943–1948.