Research Article, 2023, Volume 10, Number 1 (pp. 47–59)
http://www.jcomsec.org

Journal of Computing and Security

Development of a CGAN-Based Method for Aspect Level Text Generation: Encouragement and Punishment Factors in the Aspect Knowledge

Mohammadreza Shams [a,*], Maryam Lotfi Shahreza [a], Amir Masoud Soltani [b]

[a] *Department of Computer Engineering, Shahreza Campus, University of Isfahan, Isfahan, Iran.*
[b] *Department of Artificial Intelligence, Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran.*

ARTICLE INFO.

Article history:
Received: 5 October 2022
Revised: 8 March 2023
Accepted: 3 April 2023
Published Online: 20 June 2023

Keywords:
Deep Learning, Text Generation, Conditional Generative Adversarial Network, Aspect, Long Short-Term Memory.

ABSTRACT

Text mining systems may benefit from the use of automated text generation, especially when dealing with limited datasets and linguistic resources. Most successful text generation approaches are generic rather than aspect-specific, resulting in relatively inaccurate and similar sentences in different aspects. The present study proposes a solution to this problem by extracting aspect knowledge from relevant topics and creating the correct phrase based on the Conditional Generative Adversarial Network (CGAN) for each aspect. The proposed method produces sentences using an auxiliary dataset that cannot be distinguished from genuine sentences by the discriminator. In order to generate an auxiliary dataset, aspect-based information from datasets related to the target concept is extracted. To further improve the accuracy, the generator is encouraged or punished depending on the similarity with the training corpus. Two datasets in English and Persian are used to evaluate the performance of the proposed text generation method. The results show that adding similar aspects to the auxiliary dataset improves the quality of the generated sentences. In addition, encouragement leads to more accurate sentences, while punishment leads to more varied sentences.

1 Introduction

Natural Language Processing (NLP) is used in many areas, including translation assistants, conversation bots, sentiment analysis, and recommendation systems. Machine learning systems and software are mainly reliant on training data sets, which is not always easy to obtain, especially when dealing with language processing tasks. In addition, restrictions such as copyright or the privacy of authors should also be considered. Therefore, the automatic generation of text datasets has recently received much attention [1].

In many machine learning methods, such as deep learning, large amounts of data are required. It is sometimes necessary to compensate for a deficiency in the number of items in a specific category by using methods that produce data with specific features based on the available evidence. The same issue arises with text mining methods, and sometimes we have to generate text in a specific way before using a text mining method [2].

* Corresponding author.
Email addresses: `m.r.shams@shr.ui.ac.ir` (M. Shams), `m.lotfi@shr.ui.ac.ir` (M. Lotfi), `a.m.soltani@eng.ui.ac.ir` (A. Soltani)
https://dx.doi.org/10.22108/JCS.2023.135317.1113
ISSN: 2322-4460

We can also point to text summarization as an example, which is widely regarded as an essential topic in text mining. There are two main types of summarizations: abstractive and extractive. An abstractive method involves generating sentences that provide the best summary of the original text based on the concepts raised in the text. As in the previous case, automatic methods for text generation have a special place in this field, as well. As another example, machine translation generates text in another language based on the source text. Question Answering Systems generate textual responses to given questions. The automated chat system allows chatbots to communicate with humans using generated responses [1]. Furthermore, most successful aspect extraction and sentiment analysis methods are supervised and require a large initial training dataset; however, automatic text generation can overcome this challenge, particularly in languages with limited linguistic resources.

Automatic text generation is a subfield of NLP that uses computational linguistics and artificial intelligence to generate natural language text. Text generation falls into two major categories: random text generation and controllable text generation [3]. Random text generation requires only coherence and comprehensibility, whereas controllable text generation requires natural sentences with specific attributes including tense, sentiment, structure, grammar, and syntax.

This research focuses on controllable text generation. In this domain, language models (LMs) are used to generate texts. In order to generate texts, LMs are usually trained before being used for inference (text generation). In the training phase, LM learns the conditional probability distribution of the next token for a generated sequence of words or tokens. Then as the text generation phase, in an iterative loop, the LM's conditional probability distribution is used to select the next token. The selected tokens are also used as seeds for the next iteration. Various text generation methods are presented based on the differences in each of the mentioned phases.

The process of selecting the next token based on the conditional probability distribution is called decoding (or sampling), which is the point of difference for some text generation methods. For example, some decoding methods can be referred to greedy search (Maximization) [4], temperature sampling [5], Top-K sampling, Top-P sampling (Nucleus sampling) [5], and beam search [6]. A comprehensive overview of the different decoding methods is given in this context by Zarrieß et al. [7].

The use of language models, such as statistical and neural language models, is another important factor in the difference between text generation methods. As a successful language model, deep learning has been proposed in various forms including recurrent neural networks [8], encoder-decoder models [9], and generative adversarial networks (GANs). GANs are popular deep learning methods because they can be trained and converged quickly.

Ian Goodfellow proposed GAN in 2014 to produce new images from old ones [10], which includes a generator and a discriminator. The generator tries to generate new samples based on the original and often random set of data, while the discriminator has the task of identifying whether the new data is real or created by the system (fake). The discriminator information combined with the hidden layer information created at each stage enables the generator to improve its performance. It is repeated until the discriminator can no longer distinguish between the real and generated data.

Conditional GAN (CGAN) [11] is one of the more effective GAN extensions. In CGAN, in addition to noise as the initial input of the generator, an additional set of information is provided for it. This additional information may be in the form of class labels or any other form of the auxiliary dataset, which is an extension to the latent space to generate better samples.

Wang and Wan [12] proposed a network that generates positive and negative sentiments in English sentences using two generators and a discriminator that labels each comment with a positive or negative label. Wang and Wan [12] also use penalty-based objectives to diversify productive sentences. However, this method is limited to positive and negative statements and does not produce item-specific sentences of one aspect.

Although Conventional Generator Networks can produce generic texts, it remains challenging to develop a system capable of producing new controlled data in a particular set of aspects, with efficient and diversified results. Here, inspired by [12], we investigate the auxiliary datasets in producing better sentences and provide a CGAN-based approach to generate aspect-based sentences related to the target concept. This study identified similar aspects as the starting point for generating automatic aspect-specific phrases. To improve the accuracy of the sentences, auxiliary data with similar features are added to CGAN.

Auxiliary datasets influence sentence generation either through encouragement or punishment. As encouragement, the scores of the generated sentences that are more similar to those in the dataset will be increased, and as punishment, the scores of matched cases will be decreased. As we expected, the results indicate that encouragement improves the quality of sentences, while punishment diversifies them. The method is applied to two datasets in Persian and English to evaluate its accuracy.

This method offers the following key advantages:

- The proposed method may be able to produce more accurate and detailed statements than previous GANs that were primarily used for the production of generic texts without any knowledge about similar aspects.
- The proposed method not only generates appropriate sentences in different specific aspects but also incorporates factors of encouragement and punishment, which can affect the generated sentence quality and diversity, respectively.
- Although most of the existing methods are specific to English, the purpose of this study is to present a framework that can be used to process both Persian and English languages.

This paper is organized as follows: We review text generation methods, briefly, in Section 2. Section 3 provides a complete description of the proposed method. In Section 4, the performance evaluation of the algorithm is discussed. Finally, Section 5 summarizes the study and its results.

2 Related Works

Although GANs were originally designed to generate examples from samples in a continuous domain (e.g., image generation), their performance is increasing too quickly in discrete domains, such as text generation. There are some good studies and reviews in this domain. Aggarwal et al. [13] introduced GAN and discussed its applications in various systems. Then, they reviewed the state of GAN research in several disciplines. Kumar et al. [14] surveyed multiple aspects of GAN, including its applicability, difficulties, and evaluation criteria. In addition, they introduced state-of-the-art methods and benchmark datasets in this field.

While studying different deep learning methods for text generation, Iqbal et al. [15] discussed GAN and its problems and solutions. Saxena et al. [16] categorized various GAN approaches and also provided optimization techniques to solve related problems. Ghosh et al. [17] provided an overview of how GANs are employed, their progress, and their effectiveness. A comprehensive survey on text generation using generative adversarial networks is presented in [18].
CGAN [11] is a GAN extension that adds the class label as input to generate samples more accurately. As a result of its high efficiency, we made CGAN the main base of our method as well as one of the baseline methods for comparing results. English sentences with positive and negative polarity are produced in [12] using CGAN and a penalty function.

In [2], the authors describe a text-generating technique that can improve classifier performance for both long and short texts. The technique creates new sentences by fusing linguistic patterns with pre-trained models. A multi-scenario text generating technique based on meta-reinforcement learning is suggested in [19]. In order to generate text in a novel scenario, this approach first learns the initial parameters from numerous training tasks, then fine-tunes them in the target task.

Despite the mentioned studies in text generation, it is necessary to specify an aspect of interest to control the content of the generated text more precisely. Text content should be specified at different levels of granularity and in relation to aspects of interest. Even though aspect-based generation has a lot of potential utility, relatively little research has focused on it. This is partly due to a lack of data resources available to train and apply models to various domains. The proposed method addresses this challenge by incorporating encouragement and punishment factors into the aspect knowledge.

3 PEA-CGAN Method

On the basis of CGAN, we have developed a new method for automatic text generation. To this end, two improvements are made: encouragement and punishment are added, and sentences with similar aspects are used as initial auxiliary data. The presented model could be used for any language, but this study focuses on English and Persian. Since in the proposed method, the CGAN is enriched with the Punishment or Encouragement factor and knowledge of the Aspects, we call it PEA-CGAN in the following.

The PEA-CGAN model extracts auxiliary sentences based on the similarity of aspects. Based on the defined objective functions, the generator, and discriminator are then configured and run until convergence is achieved. The output is new sentences related to desired input aspect. Figure 1 represents this process as a workflow, with details provided below. To begin working with GAN-based methods, an initial seed of sentences is required, which is often chosen at random from a set of general sentences. This initial seed appears to influence the GAN model's ability to generate better sentences. We, therefore, provided it with a set of sentences related to the desired concept, so that instead of selecting from general sentences, it could choose from sentences that are more closely related

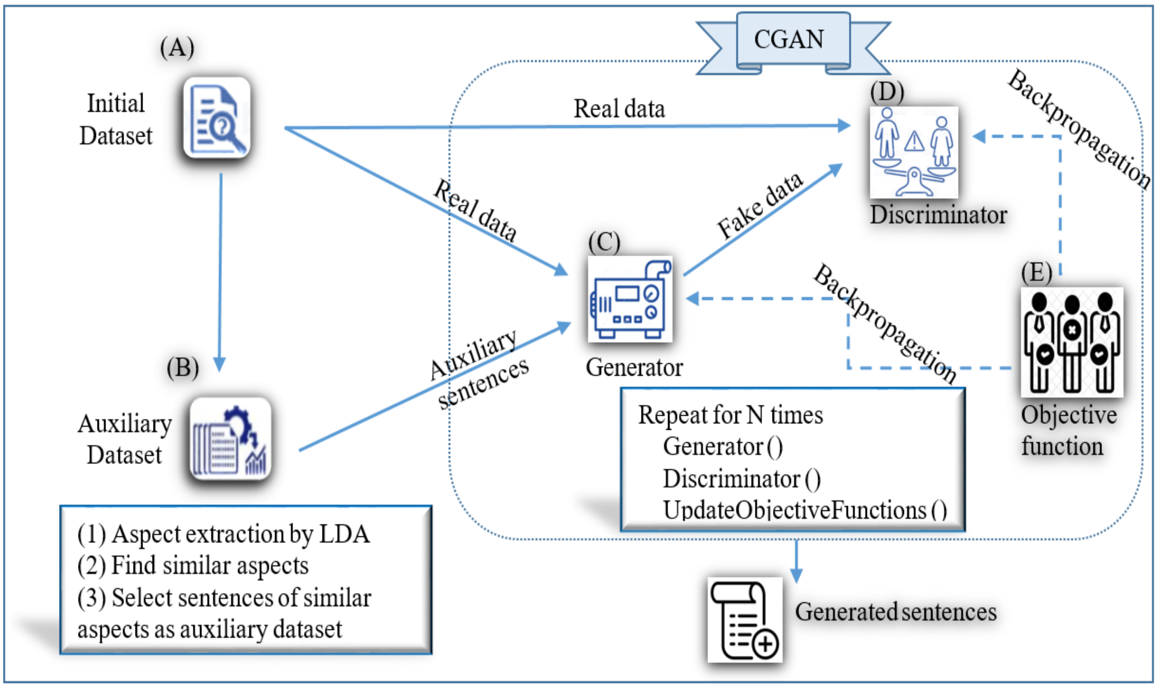

Figure 1. Workflow of the Proposed Method.
(A) Gathering the desired dataset
(B) Selecting similar sentences of other aspects and aggregating them as auxiliary datasets (explained in Section 3.1)
(C) Creating fake sentences by the generator (explained in Section 3.2)
(D) Discriminating whether the sentence is real or fake by considering its aspect (explained in Section 3.2)
(E) Calculating the objective function with encouragement or punishment factors (explained in Section 3.3)
Output: A set of newly generated sentences in each aspect.

to the concept.

Separate sentences should be created for each concept. Initially, topic modeling is performed on the sentences in the primary dataset, and the concept most closely aligned with the desired concept is chosen. Using sentences related to the closest concept, we build an initial seed we call the auxiliary dataset. Afterward, GAN is started by randomly selecting some sentences from this related set of sentences.

3.1 Extraction of Auxiliary Sentences

The proposed method uses similar aspects from other topics as a secondary dataset in order to begin with, since some concepts recur across topics, for example, CPU aspects on topics such as cell phones, tablets, and laptops. Additional topics can enrich each aspect. A similarity criterion can determine whether two aspects are similar after first identifying the various aspects. We extract the aspects of each topic based on the basic LDA method [20]. LDA is one of the most popular topic modeling methods, and its output is a set of related words that together form an aspect. Section 4 examines an example of LDA output.

Shams and Baraani-Dastjerdi [21] developed a similarity measure to identify similar aspects (Equation (1)). We determine which aspects are similar to each other based on this similarity and then rank them.

$$similarity(A_i, A_j) = \sum_{w \in A_i} \sum_{w' \in\ A_j,\ w' \neq w} Relation\,(w, w') \tag{1}$$

Where w and w represent two separate words and A_i and A_j are the i-th and j-th aspects of the topic T. Equation (1) determines the similarity of two aspects based on the co-occurrence of their words. Equation (2) calculates the $relationship(w, w')$.

$$Relation\,(w, w') = \frac{\sum_{d|(w,w') \in d} \frac{1}{|d|}}{\sum_{d|w' \in d} \frac{1}{|d|}} \tag{2}$$

Where, d is a document in topic T and $|d|$ is the length of each document. Equation (2) is based on the fact that words with similar aspects are repeated more frequently together throughout the document. On the other hand, when, w' is a more specific word, the two words are more similar. In Equation (2), the effect of document length is normalized by $\frac{1}{|d|}$.

At the end of this step, sentences related to similar aspects (based on the probability calculated in LDA) are selected as an auxiliary dataset. This auxiliary dataset and the initial sentences provide the input for the next step.

In summary, the construction of the auxiliary dataset includes the following steps:

- Identifying the aspects of each topic by LDA
- Finding similar aspects in different subjects by Equation (1)
- Extracting similar sentences as an auxiliary data set (by the probability calculated in LDA).

3.2 Text Generation

This study proposes a new method for automatically generating aspect-based sentences involving different aspects using Conditional GAN (CGAN), motivated by [12]. At first, the generator and discriminator should be initialized with appropriate values. This is done by maximum likelihood estimation (MLE), with the purpose of faster runs. MLE is a probabilistic framework that aims to resolve density estimation problems. This algorithm maximizes a likelihood function to find the best probability distribution and parameters that explain the observed data. The generator and discriminator run consecutively and repeatedly after initialization to optimize the objective function. Here we define a generator $G(S;\theta_g)$ and a discriminator $D(S;\theta_d)$, where θ_g represents the generator parameters, and θ_d represents the discriminator parameters, as well as sampling the noise z as the initial value of the generator from the P_z distribution (e.g., normal distribution).

The framework can be divided into two opposite training components: generator training and discriminator training. The generator uses Long Short-Term Memory (LSTM) [22] to generate new sentences. LSTM takes advantage of recursive layers, which exploit the influence of previous information on the current decision. It comprises multiple layers, each with a hidden state, and each cell in the layer computes the next state (h_t) based on the previous hidden state (h_{t-1}) and the embedded input x_t.The output is then sent to the discriminator. In this step, the discriminator separates the real sentences from the fake ones and returns the result to the generator. This repetition should continue until the discriminator is unable to identify the real sentences from the fictitious ones.

The main challenge lies in the fact that the generator can only generate a part of a sentence at a time, whereas the discriminator needs to detect the real sentence from the fake one, therefore sampling is used to complete the sentence. Unlike Wang and Wan [12], who used a general dataset, in this procedure, the remaining words of the sentence are completed randomly based on the probabilities within the auxiliary dataset. This completes the sentence and provides a more accurate discriminator decision. Each iteration between the generator and discriminator updates the objective function values, which are used to weight the parameters in the following iteration.

3.3 Objective Function

In CGAN, the Minimax loss function is the main objective function (Equation (3)). As the generator aims to minimize it, the discriminator seeks to maximize it.

$$
\begin{aligned}
&min_G max_D V(D,G) = \\
&\qquad E_{x\sim P_{data(x)}}[\log D(x|\ y)] + \\
&\qquad E_{z\sim P_{z(z)}}[\log(1 - D(G(z|y)))]
\end{aligned} \tag{3}
$$

where,

- x represents the real data, y represents auxiliary data, and z represents noise data.
- $D(x|y)$ is the discriminator's estimate of the probability that real sample instance x is real.
- The E symbol is used for the expected value over all related data instances.
- $G(z|y)$ is the output of the generator.

According to Equation (3), the most important factor of the text generation process is $G(z|y)$ (the output of the generator) hence it is necessary to choose the most appropriate objective function so that the sentences generated are both correct and diverse.

In order to accomplish this, two different objective functions are proposed and compared: one with a punishment factor and the other with an encouragement factor. We claim that punishment makes sentences more diverse, while encouragement makes them more accurate. The proposed generator objective function is $J_G(S)$ in Equation (4).

$$
\begin{aligned}
&J_G(S) = \\
&\qquad \begin{array}{ll}
E_{x\sim P_g}[-log(D(S|y,\ \theta_d))] & CGAN \\
E_{x\sim P_g}[-log((D(S|\ y,\theta_d)\,G(S|\ \theta_g)] & WithEncouragement \\
E_{x\sim P_g}[-log((D(S|\ y,\theta_d)\,(1 - G(S|\ \theta_g))] & WithPunishment
\end{array}
\end{aligned} \tag{4}
$$

where,

- S: Generated sentence by the generator
- $D(S|y,\theta_d)$: The probability of real label by the discriminator for the sentence generated using the auxiliary dataset

```xml
<Sentence id="0">
    <Tokens>This product is just as described and works perfectly !</Tokens>
    <POS>DT NN VBZ RB IN VBN CC VBZ RB .</POS>
    <Lemma>this product be just as describe and work perfectly !</Lemma>
</Sentence>
```

Figure 2. A Sample of English Dataset. The sentence, Part-of-speech, and lemmatization of the sentence words are expressed in labels "Tokens", "POS", and "Lemma", respectively. It should be noted that only the basic sentences (Tokens) are used in this manuscript.

- $G(S|\theta_g)$: The probability of generation sentence S.

Both encouragement and punishment can be used to create a new sentence based on Equation (4). In the case of encouragement, the similarity of the resulting phrase to the auxiliary dataset is viewed positively, resulting in faster convergence of the proposed method. Instead, in the punishment case, when sentences are penalized for their proximity to the auxiliary dataset, they become more unique.

An LSTM structure is used to examine how likely it is that a sentence will be made ($G(S|\theta_g)$). In this structure, the probability of each word being used is predicted based on the previous state and the words that came before it in the sentence. As shown in Equation (5), the final generation probability of each sentence is based on the conditional probability that each word will happen in the previous state and word.

$$G(S|\theta_g) = \sum_{t=1}^{|s|} G\left(S_{t+1}|S_t,\ \theta_g\right) = \sum_{t=2}^{|s|} softmax(LSTM_{\theta_g}\left(S_{t-1}, h_{t-1}\right) \tag{5}$$

In the suggested strategy, $D(S|y, \theta_d)$ is the likelihood of being able to distinguish between real and fake phrases, which is calculated by sampling from the auxiliary dataset. To do so, the discriminator evaluates the likelihood that the phrase is real in each cycle by sampling the first t tokens from the model and the remaining ($|S| - t$) tokens from the auxiliary dataset (Equation (6)).

$$D\left(S|y,\ \theta_d\right)) = D\left(S_{1:t}^{t+1:|S|} \mid \theta_d\right) \tag{6}$$

Where, $S_{1:t}^{t+1:|s|}$ is a sentence whose t first words is determined by the model and the rest of the words are determined by Monte Carlo sampling. We'll look at how effectively the suggested approach and the auxiliary dataset operate at creating sentences in the following section, as well as how encouraging and punishing variables work.

4 Experimental Results

The first subsection describes how to extract the required data and presents the dataset used for evaluating the proposed method. In the second subsection, methods used for comparison are introduced. Subsections 4.3 and 4.4 present the evaluation results qualitatively and statistically, respectively.

The proposed evaluation plan comprises two elements:

- Intuitive evaluation: Three items are used to determine the quality that humans perceive in the generated sentences: similar aspect extraction, generated sentence samples, and aspects enriched with generated sentences.
- Statistical evaluation: Three parameters are used to evaluate the generated sentences: perplexity, topic coherence, and diversity.

This section evaluates not only the quality of generated sentences but also the effect of sentences on improving the aspect extraction process. The aspect extraction process is repeated for this purpose by adding generated sentences to the initial dataset, and the results are investigated intuitively and statistically.

4.1 Dataset

Two datasets are used to evaluate the proposed method in Persian and English. The first dataset includes 13 different topics in the field of electronic devices collected from online businesses in Persian. Over 20,000 documents promoting various electronic equipment projects have been collected and will evaluate the proposed method [23]. The second dataset contains 50 different electronic devices, each containing 1000 English documents [24]. As both datasets are based on the electronic device concept, they can be compared in similar ways. Figure 2 represents a sentence from this collection.

The proposed method attempts to determine and utilize similar aspects to increase accuracy. To illustrate how similar aspects can be found in each dataset, think of the "CPU" part of a laptop, which is also found with comparable terminology on other devices like mobile phones, tablets, and so on.

Statistical criteria and qualitative evaluations extracted from the two datasets are used to evaluate the proposed methodology. The laptop's screen, storage, battery, and memory are all examined throughout the examination. The four aspects were selected manually, but since no special processing is applied to the aspects, the generality of the method is not violated and any other aspect can be used in the model. The aspects selected for the research can be found in both databases to the greatest extent feasible, thus making comparisons fair and making it easier for the reader to understand.

Separate Persian and English datasets are evaluated, and the results are summarized in the following subsections.

4.2 Baselines

The obtained results of the PEA-CGAN are compared with five state-of-the-art baselines:

- LDA [20]: LDA is a topic model whose output is aspects related to each topic. The LDA is used to evaluate the proposed method in aspect extraction.
- CGAN [11]: Conditional GAN is one of the successful extensions of GAN that uses an auxiliary dataset as conditional variables to conduct the generation of new samples. In this manuscript, a set of "notebook" sentences are imported into CGAN as an auxiliary dataset to generate sentence examples related to each aspect. It should be mentioned that all phases of the CGAN technique are implemented in the same manner as a PEA-CGAN; the only difference is that the generator objective function is calculated without encouragement or punishment (Equation (4)).
- SentiGAN [12]: SentiGAN generates texts with different sentiment labels (positive and negative). The generator also uses a penalty-based objective to diversify the sentences produced. This method requires three datasets for training: positive, negative, and general auxiliary sentence sets. To use this method in text generation in a specific aspect, the desired aspect (from the four aspects screen, storage, battery, and memory) is taken as the positive dataset, whereas the other three aspects are considered negative. General auxiliary data, similar to CGAN, consider the laptop without considering its aspect.
- BERT: BERT [25] is a language representation model that has performed surprisingly well in a variety of language understanding benchmarks. BERT is trained as a bidirectional Masked Language Model (MLM). Rather than predicting every next token, MLM randomly masks a percentage of input tokens, and word tokens are predicted based on the simultaneous use of left and right word contexts. BERT is evaluated in automatic text generation by randomly masking 15% of each sentence's words and replacing each token with the output of a pre-trained model.
- ParsBERT: ParsBERT [26] is a monolingual BERT for Persian that performs well in Persian NLP tasks and is lighter than multilingual BERT. PARSBERT, like BERT, is used to evaluate the proposed method. After masking 15% of the words in each sentence, the alternative word is chosen from the pre-trained model.

Table 1. Selected Topics to Extract Similar Aspects.

Aspects of Laptop topic	Topics with similar aspects: English dataset	Topics with similar aspects: Persian dataset
Screen	Graphic Card Cellphone	PC Cellphone
Storage	Home Theater System Video Recorder	PC Network Equipment
Battery	Video Recorder MP3 Player	Tablet and E-Reader Digital Camera
Memory	Cellphone Camcorder	Tablet and E-Reader Cellphone

Table 2. List of words for storage and battery aspects in laptop and similar topics. The related words are in green.

Battery			Storage		
MP3 Player	Video Recorder	Laptop	Home Theater System	Video Recorder	Laptop
battery	battery	battery	dvd	hard	drive
hour	hour	hour	player	drive	hard
zune	life	life	cd	recording	dvd
life	aaa	charge	disc	computer	case
long	long	minute	changer	dvd	cd
hd	mode	full	vcr	movie	netbook
charge	rechargeable	long	space	space	notebook
full	capacity	original	tray	disk	box
aaa	charge	cell	tuner	part	external
app	alkaline	longer	playback	series	player

4.3 Intuitive Evaluation

To appropriately evaluate the proposed method, it is first intuitively investigated and then confirmed by statistical criteria.

4.3.1 Similar Aspect Extraction

As mentioned earlier, the proposed method improves sentence generation performance using sentences with similar aspects. The most similar aspects are found throughout the entire dataset (all 50 and 13 topics in the English and Persian dataset, respectively). The aspects that are most similar to the desired concept are selected and used. Table 1 lists the two topics with the highest similarity to each desired aspect (according to Equation (1)).

As predicted, we can uncover aspects that are similar across a variety of topics, and these could serve as auxiliary datasets. Table 1 shows that the memory aspect of laptops appears in various topics such as cellphones, camcorders, and tablets, from which phrases may be extracted to generate auxiliary datasets. Table 2 lists the words used in similar aspects in both topics.

Table 2 shows that there are many semantic similarities between aspects that are related across multiple topics. Taking advantage of this knowledge can improve productive phrases. Due to the similarity between the words of similar aspects, the sentences associated with these aspects will also be related to each other and can be used as auxiliary sentences for automatic sentence generation.

Please note that the results of intuitive evaluations are only given in English. This is so that the reader may comprehend them better. But in the comparison criteria in the next section, both Persian and English are taken into account.

4.3.2 Samples of Generated Sentences

We compare the sentences derived from the proposed method with those generated from SentiGAN and CGAN to assess their effectiveness in extracting similar aspects and using them as auxiliary data. In Table 3, we present three baselines as well as examples of sentences generated by the proposed method in both encouragement and punishment procedures.

The way the punishment factor is expressed through the sentences, which can be observed visually, will lead to a great deal of variety. On the other hand, the encouragement technique produces more efficient phrases with higher linguistic quality.

The sentences produced by BERT are mostly incomprehensible and contain repetitive and generic words. This poor performance is due to the use of a generic pre-trained model. SentiGan and CGAN both use general auxiliary datasets, so their results are inferior to PEA-CGAN. The proposed PEA-CGAN uses similar aspects of auxiliary datasets to produce sentences

Table 3. The Resulting Sentences from PEA-CGAN, SentiGAN, BERT, and CGAN for the "Battery" Aspect.

Mode of generator	Generated sentences
SentiGAN	so I use a <UNK>laptop rest to place the cooler on top to give it the space it need a dvd movie when watch the memory card. Great for go through airport security <UNK>, but need more storage compartment.
CGAN	the red and silver along with the tablet style keyboard and the red backlight just make it look the Amazon review do not get the memory information right but everything else be right on <UNK>.
BERT	i both the initial charge guide failed and the battery did not charge at first, but at 1 %. in battery do it last 8 hour like me but it be more than normal computer in like.
PEA-CGAN with punishment	it be <UNK>that some owner report good battery life on last a lower voltage, so if the battery last drain set also, my battery only last 2 minute and the battery last just what apple, with waifus on, surf web, watch.
PEA-CGAN with encouragement	the screen go off after a set amount of time be nice, because it save up on battery power. it offer lot more function and improve battery life as well.

that are more closely aligned with the topics people desire to discuss. For example, as shown in Table 3, statements about other aspects (storage, keyboard, etc.) are incorrectly generated as battery aspects by CGAN and SentiGAN.

4.3.3 Enriched Aspects by Generated Sentences

Another evaluation investigates the influence of generated sentences on aspect extraction, so we add the generated sentences to the initial dataset and repeat the aspect extraction by LDA. Based on this test, Table 4 shows the results for various methods. The related or unrelated (black or red) label was chosen based on the majority opinion of three laptop domain experts.

Table 4 shows that many inconsistencies are found in the words retrieved by basic LDA. It can be ar-

Table 4. Using the sentences generated by PEA-CGAN, SentiGAN, BERT, and CGAN to improve the accuracy of aspect extraction (Errors are in red).

Method	Label of Aspect	Words	Number of errors
Basic LDA	**Screen**	screen, keyboard, mouse, key, wireless, monitor, mode, size, brightness, light	4
	Storage	drive, hard, dvd, case, cd, netbook, notebook, box, external, player	3
	Battery	battery, hour, life, charge, minute, full, long, original, offer, longer	1
	Memory	memory, ram, card, video, game, gb, slot, upgrade, program, processor	4
LDA+BERT	**Screen**	screen, look, like, key, full, size, back, notebook, mode, keyboard	5
	Storage	hard, drive, cool, dvd, case, product, service, old, helpful, graphic	4
	Battery	hour, life, battery, last, price, save, whole, set, charge, power	3
	Memory	memory, time, install, video, new, card, crash, play, system, sd	4
LDA+CGAN	**Screen**	screen, keyboard, angle, brightness, view, picture, experience, turn, tablet, star	4
	Storage	dvd, read, box, trouble, cd, desk, space, spend, adapter, burner	4
	Battery	battery, year, explode, reference, mysterious, analysis, gaming, search, student, bring	8
	Memory	Memory, desktop, crash, card, faster, upgrade, cooler, graphic, send, dell	4
LDA+SentiGAN	**Screen**	screen, picture, play, video, product, graphic, card, dell, music, option	3
	Storage	drive, file, copying, inch, store, connection, bad, internet, fact, memory	4
	Battery	battery, charge, power, buy, review, read, set, longer, start, supply	4
	Memory	memory, crash, install, remove, window, boot, slot, restore, cost, upgrade	4
LDA+ PEA-CGAN with punishment	**Screen**	screen, display, resolution, inch, pro, view, people, bit, notebook, brightness	3
	Storage	drive, usb, mode, flash, plug, include, digital, load, port, cable	1
	Battery	battery, charge, lot, function, offer, rechargeable, improve, player, month, user	5
	Memory	memory, card, sd, hold, stick, slot, purchase, gb, tape, space	1
LDA+ PEA-CGAN with encouragement	**Screen**	screen, love, touch, inch, year, resolution, larger, map, replace, design	3
	Storage	drive, usb, recording, storage, portable, flash, port, longer, datum, load	0
	Battery	battery, save, plug, change, aaa, charge, provide, setting, cell, die	0
	Memory	memory, card, laptop, sd, hold, stick, crash, slot, gb, recommend	1

gued that words such as “keyboard”, “mouse” and “key” for the screen aspect or “netbook” and “laptop” for the storage aspect may not match with other words. When compared to a basic LDA, the addition of general knowledge through SentiGAN, CGAN, and BERT increases the aspect inconsistency. However, the proposed approach clearly improves the aspects due to the insertion of productive sentences.

The punishment technique introduces more broad words for each aspect, such as “pro”, “view”, and “people”, but the encouragement method identifies specific words for each aspect more accurately. Note that word tags are assigned manually to each aspect and exist only for clarification purposes. The next subsection discusses the statistical evaluation of the proposed method.

4.4 Statistical Evaluation

To conduct statistical analyses, the effectiveness of the proposed methods, as well as the punishment and punishment methods, is first measured in terms of topic coherence and perplexity, followed by the diversity of the sentences produced.

4.4.1 Perplexity

According to Equation (7), Perplexity is used to predict the appropriateness of a probability distribution or probabilistic model.

$$Perplexity = 2^{-\frac{1}{N}log_2 P(w_1,w_2,\,\ldots,w_N)} \qquad (7)$$

Where $P(w_1, w_2, \ldots, w_N)$ is an n-gram model:

Table 5. The Perplexity Criterion for the Generated Sentences by PEA-CGAN, SentiGAN, BERT, PARSBERT, and CGAN.

	Aspect	SentiGAN	CGAN	BERT	PEA-CGAN	
					Punishment	Encouragement
English Dataset	Screen	144.1	131.8	92.3	61.3	**58.4**
	Storage	159.4	153.1	113.9	65.4	**62.6**
	Battery	115.5	109.3	81.3	40.9	**40.8**
	Memory	162.1	151.9	156	88.5	**87.5**
	Aspect	**SentiGAN**	**CGAN**	**PARSBERT**	**PEA-CGAN**	
					Punishment	**Encouragement**
Persian Dataset	Screen	151.6	150.1	191.5	165.6	**146.5**
	Storage	156.8	158.7	173.6	182.4	**152.2**
	Battery	183.8	182.7	195.2	125.4	**121.6**
	Memory	158.2	158.8	166.7	124.4	**124.2**

Table 6. Topic coherence of each model on the English and Persian dataset.

	Aspect	LDA	SentiGAN	CGAN	BERT	PEA-CGAN	
						Punishment	Encouragement
English Dataset	Screen	-261.5	-257.7	-243.7	-199.2	-288.9	**-175.9**
	Storage	-269.7	**-204.7**	-247.4	-217.8	-236.8	-215.9
	Battery	-267.9	-182.8	-216.4	-178.4	-170	**-167.9**
	Memory	-245.5	-265.9	-212.1	-220.1	-216.7	**-211.5**
	Aspect	**LDA**	**SentiGAN**	**CGAN**	**PARSBERT**	**PEA-CGAN**	
						Punishment	**Encouragement**
Persian Dataset	Screen	-174.7	-127.1	-124.6	-174.8	**-112.2**	-149.8
	Storage	-243.5	-128.5	-139.9	-201.1	-145.8	**-127.9**
	Battery	-124.6	-183.3	-164.2	-188.5	-166.2	**-122.7**
	Memory	-216.9	-218.8	**-158.2**	-208.7	-198.4	-207.8

Table 7. The average diversity of generated sentences by each model on the English and Persian dataset.

English Dataset	SentiGAN	CGAN	BERT	PEA-CGAN	
				Punishment	Encouragement
	0.703	0.687	0.692	0.698	0.673
Persian Dataset	**SentiGAN**	**CGAN**	**PARSBERT**	**PEA-CGAN**	
				Punishment	**Encouragement**
	0.689	0.0661	0.718	0.659	0.644

$$p(w_1, w_2, w_3, \ldots, w_n) = p\left(w_1\right) * p\left(w_2|w_1\right) * p\left(w_3|w_1, w_2\right) \ldots * p\left(w_n—w_{n-2}, w_{n-1}\right) \quad (8)$$

In this study, the occurrence of each word depends on only two previous words (Trigram). The Perplexity for the mentioned objective functions and baseline methods can be seen in Table 5.

According to the perplexity equation, since the perplexity is related to the inverse of the geometric mean, it can be concluded that the lower value gives a better result. The evaluation results show that the proposed PEA-CGAN outperforms BERT and PARSBERT. Language modeling is directly related to sentence generation (given the previous words in the sentence, what is the next word). Because of their bidirectionality, BERT and PARSBERT cannot be used as perfect sentence generators. Furthermore, the PEA-CGAN with encouragement factor generates output with higher accuracy than the other baseline methods.

4.4.2 Topic Coherence

The topic coherence criterion is used to perform the next evaluation. In one sense, this refers to the link between the words. When the words used in one aspect are more repeated together and also the aspect words are more specific, the extracted aspect becomes more coherent. Equation (9) [23] is used to calculate this criterion.

$$Topic\ Coherence = \sum_{i<j} \log_{10} \frac{CoDF\,(w_i.\ w_j) + 1}{DF\,(w_j)} \quad (9)$$

Where, $CoDF(w_i.w_j)$ is the co-occurrence of two words in different documents and $DF(w)$ is the document frequency of word w. Table 6 evaluates the proposed method based on this criterion.

Table 6 illustrates how accurate supporting knowledge can greatly improve coherence in all areas. As expected, coherence improved more in the encouraging function mode than in the other cases.

According to Table 6, the proposed method has the highest coherency, performed in five of the eight tests (four tests in English and four tests in Persian). The investigation of three other cases reveals that the results are due to a lack of proper diversity in the dataset.

It should also be noted that a method cannot be evaluated solely based on topic coherence [23]. The examination of all of the evaluation criteria demonstrates that the proposed method outperforms other methods.

4.4.3 Diversity

Using Equation (10) and the Jaccard similarity [12], the present research calculates the diversity of sentences produced by each aspect.

$$diversity(A) = \sum_{S_i \in A} \left(1 - max \left\{ \frac{|S_i \cap S_j|}{|S_i| + |S_j| - |S_i \cap S_j|} \right\}_{j=1}^{j=|S|,\ i \neq j} \right) \quad (10)$$

Where $|S_i|$ is the length of sentence i and $|S_i \cap S_j|$ is the number of common words between two independent sentences i and j. According to Equation (10), the diversity and number of repetitive words in two sentences are inversely related. Table 7 evaluates the proposed method based on the diversity metric.

Table 7 shows that the proposed method produces more diverse sentences in the punishment mode than in the encouragement mode. This diversity is due to the penalty imposed in the punishment factor for approaching the auxiliary datasets.

The general knowledge added to BERT, SentiGAN, and CGAN increases diversity. But, as mentioned in previous sections, the general dataset significantly reduces the quality of the generated sentences and their relation to the desired aspects.

5 Conclusions

This paper presents a CGAN-based method for automatic sentence generation that enriches various aspects of a topic. The proposed approach uses similar aspects in different topics as auxiliary datasets for CGAN, and these data improve the output sentences' accuracy. The CGAN algorithm incorporates encouragement and punishment methods to implement the objective function. In the encouragement technique, if the generated sentence is closer to the auxiliary dataset, the impact of the encouragement factor in the objective function is increased, and as a result, the method leads to more precise sentences that will be closer to the auxiliary dataset. Alternatively, the punishment method reduces the accuracy of the sentences while increasing their variety by applying penalty factors to the auxiliary dataset. The results of intuitive and numerical evaluations confirm the accuracy of the claim.

References

[1] W. Yu, Ch. Zhu, Z. Li, Zh. Hu, Q. Wang, H. Ji, and M. Jiang. A survey of knowledge-enhanced text generation. *ACM Computing Surveys*, 54:1–38, 1 2022. ISSN 0360-0300. doi:10.1145/3512467. URL `https://dl.acm.org/doi/10.1145/3512467`.

[2] M. Bayer, MA. Kaufhold, B. Buchhold, M. Keller, J. Dallmeyer, and Ch. Reuter. Data augmentation in natural language processing: a novel text generation approach for long and short text classifiers. *International Journal of Machine Learning and Cybernetics*, 14:135–150, 1 2023. ISSN 1868-8071. doi:10.1007/s13042-022-01553-3.

[3] Sh. Prabhumoye, A. W. Black, and R. Salakhutdinov. Exploring controllable text generation techniques. pages 1–14. International Committee on Computational Linguistics, 2020. doi:10.18653/v1/2020.coling-main.1. URL `https://www.aclweb.org/anthology/2020.coling-main.1`.

[4] A. BELZ. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14:431–455, 10 2008. ISSN

1351-3249. doi:10.1017/S1351324907004664. URL https://www.cambridge.org/core/product/identifier/S1351324907004664/type/journal_article.

[5] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi. The curious case of neural text degeneration. *CEUR Workshop Proceedings*, 2540, 2019. ISSN 16130073. URL http://arxiv.org/abs/1904.09751.

[6] W. Kool, H. van Hoof, and M. Welling. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. 3 2019. URL http://arxiv.org/abs/1903.06059.

[7] S. Zarrie, H. Voigt, and S. Schüz. Decoding methods in neural language generation: A survey. *Information*, 12:355, 8 2021. ISSN 2078-2489. doi:10.3390/info12090355. URL https://www.mdpi.com/2078-2489/12/9/355.

[8] H. Hayashi. Neural aspect-based text generation, dissertation, carnegie mellon university. Technical report, Dissertation, Carnegie Mellon University, 2021.

[9] P. Sun, L. Wu, K. Zhang, Y. Su, and M. Wang. An unsupervised aspect-aware recommendation model with explanation text generation. *ACM Transactions on Information Systems*, 40:1–29, 7 2022. ISSN 1046-8188. doi:10.1145/3483611. URL https://dl.acm.org/doi/10.1145/3483611.

[10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, Sh. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. 6 2014. URL http://arxiv.org/abs/1406.2661.

[11] M. Mirza and S. Osindero. Conditional generative adversarial nets. 11 2014. URL http://arxiv.org/abs/1411.1784.

[12] K. Wang and X. Wan. Sentigan: Generating sentimental texts via mixture adversarial networks. pages 4446–4452. International Joint Conferences on Artificial Intelligence Organization, 7 2018. ISBN 9780999241127. doi:10.24963/ijcai.2018/618. URL https://www.ijcai.org/proceedings/2018/618.

[13] A. Aggarwal, M. Mittal, and G. Battineni. Generative adversarial network: An overview of theory and applications. *International Journal of Information Management Data Insights*, 1:100004, 4 2021. ISSN 26670968. doi:10.1016/j.jjimei.2020.100004. URL https://linkinghub.elsevier.com/retrieve/pii/S2667096820300045.

[14] M. R. Pavan Kumar and P. Jayagopal. Generative adversarial networks: A survey on applications and challenges. *International Journal of Multimedia Information Retrieval*, 10:1–24, 3 2021. ISSN 2192-6611. doi:10.1007/s13735-020-00196-w. URL https://link.springer.com/10.1007/s13735-020-00196-w.

[15] T. Iqbal and Sh. Qureshi. The survey: Text generation models in deep learning. *Journal of King Saud University - Computer and Information Sciences*, 34:2515–2528, 4 2020. ISSN 13191578. doi:10.1016/j.jksuci.2020.04.001. URL https://linkinghub.elsevier.com/retrieve/pii/S1319157820303360.

[16] D. Saxena and J. Cao. Generative adversarial networks (gans): Challenges, solutions, and future directions. *ACM Computing Surveys*, 54:1–42, 6 2021. ISSN 0360-0300. doi:10.1145/3446374. URL https://dl.acm.org/doi/10.1145/3446374.

[17] B. Ghosh, I. K. Dutta, M. Totaro, and M. Bayoumi. A survey on the progression and performance of generative adversarial networks. pages 1–8. IEEE, 7 2020. ISBN 978-1-7281-6851-7. doi:10.1109/ICCCNT49239.2020.9225510. URL https://ieeexplore.ieee.org/document/9225510/.

[18] G. H. de Rosa and J. P. Papa. A survey on text generation using generative adversarial networks. *Pattern Recognition*, 119:108098, 11 2021. ISSN 00313203. doi:10.1016/j.patcog.2021.108098. URL https://linkinghub.elsevier.com/retrieve/pii/S0031320321002855.

[19] T. Zhao, G. Li, Y. Song, Y. Wang, Y. Chen, and J. Yang. A multi-scenario text generation method based on meta reinforcement learning. *Pattern Recognition Letters*, 165:47–54, 1 2023. ISSN 01678655. doi:10.1016/j.patrec.2022.11.031.

[20] D. M. Blei, A. Y. Ng, and M. I Jordan. Latent dirichlet allocation. *Journal of machine learning research*, 3:993–1022, 2003. URL https://dl.acm.org/doi/abs/10.5555/944919.944937.

[21] M. Shams and A. Baraani-Dastjerdi. Enriched lda (elda): Combination of latent dirichlet allocation with word co-occurrence analysis for aspect extraction. *Expert Systems with Applications*, 80:136–146, 9 2017. ISSN 09574174. doi:10.1016/j.eswa.2017.02.038. URL https://linkinghub.elsevier.com/retrieve/pii/S095741741730129X.

[22] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 11 1997. ISSN 0899-7667. doi:10.1162/neco.1997.9.8.1735. URL https://direct.mit.edu/neco/article/9/8/1735-1780/6109.

[23] M. Shams, N. Khoshavi, and A. Baraani-Dastjerdi. Lisa: Language-independent method for aspect-based sentiment analysis. *IEEE*

Access, 8:31034–31044, 2020. ISSN 2169-3536. doi:10.1109/ACCESS.2020.2973587. URL https://ieeexplore.ieee.org/document/8995576/.

[24] Zh. Chen and B. Liu. Topic modeling using topics from many domains, lifelong learning and big data. volume 32, pages 703–711. Proceedings of the 31st International Conference on Machine Learning - ICML '14, 2014.

[25] J. Devlin, M. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. pages 4171–4186. Association for Computational Linguistics, 10 2019. doi:10.18653/v1/N19-1423. URL https://arxiv.org/abs/1810.04805http://aclweb.org/anthology/N19-1423.

[26] M. Farahani, M. Gharachorloo, M. Farahani, and M. Manthouri. Parsbert: Transformer-based model for persian language understanding. *Neural Processing Letters*, 53:3831–3847, 12 2021. ISSN 1370-4621. doi:10.1007/s11063-021-10528-4. URL https://link.springer.com/10.1007/s11063-021-10528-4.

Mohammadreza Shams received the B.S. degree from the Isfahan University of Technology, in 2008, the M.S. degree from the University of Tehran, in 2012, and the Ph.D. degree from the University of Isfahan, Iran, in 2017, all in computer engineering. He is currently an Assistant Professor with the Computer Engineering Department, University of Isfahan (Shahreza Campus). His research interests include data mining, text mining, and deep learning.

Maryam Lotfi Shahreza received her Master degree in Computer Engineering (software) at Isfahan University of Technology in 2006. She received her Bachelor degree in Computer Engineering (Algorithm and computational sciences) at University of Tehran in 2009. She received her PhD of Computer Engineering (software) at Isfahan University of Technology in 2018. Her research interests include issues related to Data Sciences, Text mining, Complex Networks, Bioinformatics and Computational Biology, especially Systems Biology and Pharmacology Sciences.

Amir Masoud Soltani received his B.Sc. in computer engineering from the University of Isfahan (Shahreza Campus) in 2020. He is currently pursuing a master's degree in computer engineering with artificial intelligence specialty at the University of Isfahan. His research interests are Geometric Deep Learning, Self-supervised Learning, and Multi-modal Learning.