algorithms

Article

Learning Representations of Natural Language Texts with Generative Adversarial Networks at Document, Sentence, and Aspect Level

Aggeliki Vlachostergiou [1,*], George Caridakis [1,2], Phivos Mylonas [1,3] and Andreas Stafylopatis [1]

[1] Intelligent Systems Content and Interaction Laboratory, National Technical University of Athens (NTUA) 15780, Greece; gcari@image.ntua.gr (G.C.); fmylonas@image.ntua.gr (P.M.); andreas@cs.ntua.gr (A.S.)
[2] Department of Cultural Technology and Communication, University of the Aegean 81100, Greece
[3] Department of Informatics, Ionian University, 49100 Corfu, Greece
* Correspondence: aggelikivl@image.ntua.gr; Tel.: +30-210-772-4052

Received: 27 August 2018; Accepted: 19 October 2018; Published: 22 October 2018

Abstract: The ability to learn robust, resizable feature representations from unlabeled data has potential applications in a wide variety of machine learning tasks. One way to create such representations is to train deep generative models that can learn to capture the complex distribution of real-world data. Generative adversarial network (GAN) approaches have shown impressive results in producing generative models of images, but relatively little work has been done on evaluating the performance of these methods for the learning representation of natural language, both in supervised and unsupervised settings at the document, sentence, and aspect level. Extensive research validation experiments were performed by leveraging the 20 Newsgroups corpus, the Movie Review (MR) Dataset, and the Finegrained Sentiment Dataset (FSD). Our experimental analysis suggests that GANs can successfully learn representations of natural language texts at all three aforementioned levels.

Keywords: natural language texts; representation learning; deep learning; generative adversarial networks (GANs); adversarial training; document; sentence; aspect-level text analysis; information retrieval

1. Introduction

The performance of machine learning (ML) methods is heavily dependent on the choice of data or feature representation to which they are applied. For that reason, much of the actual effort in deploying ML algorithms goes into the design of preprocessing the pipelines and data transformations that result in a representation of the data that can support effective ML. Such feature engineering is important but labor-intensive, which highlights the weakness of current learning algorithms. Even though there are a large variety of approaches to representation learning in general, the underlying concept is to learn some set of features from data, and then use these features to solve, for example, a separate (possibly unrelated) task for which we have a large number of labeled examples. As a result, the emergence of large-scale datasets, such as ImageNet [1], which contains 14,197,122 manually labeled images, has allowed the wider-spread use and popularity of convolutional neural networks (CNNs) even in the unrelated task of medical imaging. Currently, the majority of existing classifiers cannot perform as expected when the size of the training dataset is small. Constructing a large labeled dataset, however, is time-consuming and usually requires domain knowledge, making it even more costly. Therefore, there is a gap between the potential benefits of having a large dataset and the difficulty in obtaining labeled data. In order to expand the scope and ease of applicability of ML, it would be highly desirable to make learning algorithms less dependent on feature engineering, so that novel applications can be constructed faster. This could be possible by learning representations (LRs) of the data that make it easier to extract useful information when building classifiers or other predictors. A good representation is one that is also useful as input to a supervised predictor. Among the various methods of LR, this paper focuses on deep learning methods: those that are formed by the composition of multiple nonlinear transformations, with the goal of yielding more abstract—and, ultimately, more useful—representations.

Among the effective approaches that have emerged to train deep generative models, the one that is based on the variational autoencoder (VAE) [2,3] and the approach that uses generative adversarial networks (GANs) have dominated in recent years. In the former one, the observed data x is assumed to be generated from a set of stochastic latent variables z. The VAE introduces an inference network (implemented using a deep neural network) to approximate the intractable distributions over z, and then maximizes a lower bound on the log-likelihood of $p(x)$. The latter approach uses GANs [4]. In the original GAN formulation, a generator deep neural network learns to map samples from an arbitrary distribution to the observed data distribution. A second deep neural network called the discriminator is trained to distinguish between samples from the empirical distribution and samples that are produced by the generator. The generator is trained to create samples that will fool the discriminator, and so an adversarial game is played between the two networks, converging on a saddle point that is a local minimum for the discriminator and a local maximum for the generator. Both VAE and GAN approaches have shown impressive results in producing generative models of images [5,6], but relatively little work has been done on evaluating the performance of these models for learning representations of natural language. GANs for natural language processing (NLP) are considered powerful methods, as they deal with generating sentences: specifically, they produce sentences with certain characteristics (sentiment and questions) and take advantage of the unsupervised nature of these deep neural network (DNN) models. One reason that GANs cannot be directly applied to natural language is the fact that the space in which sentences are present is not continuous and therefore not differentiable. On the contrary, text is represented atomically in terms of discrete tokens (like "man", "girl", etc). So, when we want to update the generated sentence slightly according to the discriminator's behavior, we may not get a sentence. In the computer vision (CV) domain, the output of the generator is an image (a matrix consisting of real valued numbers) which can undergo small updates to make it more difficult for the discriminator to differentiate between the real and fake image. Recently, however, VAEs have been used successfully to create language models [7], to model documents, and to perform question answering [8]. This paper attempts to shed light on whether GANs can be used to learn representations of natural language in an unsupervised setting.

Particularly, in our work, we formulate the ML problem as follows. We propose a novel extension of GANs that replaces the traditional binary classifier discriminator with one that assigns a scalar energy to each point in the generator's output domain. The discriminator minimizes the hinge loss function used for training "maximum-margin" classifiers, while the generator attempts to generate samples with low energy under the discriminator. We show that a Nash equilibrium [9] under these conditions yields a generator that matches the data distribution (assuming infinite capacity). We conducted experiments with the discriminator in the form of a denoising autoencoder (DAE), optionally including a regularizer that penalizes generated samples having a high cosine similarity to other samples in the mini-batch. Our proposed neural network architecture is based on a variation of the recently proposed energy-based GAN [10] that has been proven to be suitable for the task of generating high-resolution MNIST digit images [11], providing a qualitative evaluation of the learned representations. Additionally, we decided to replace the standard probabilistic GAN that we cast into the energy model (using Gibbs distributions [12]), as has been proposed by the work of Kim and Bengio [13], and we decided to present the same Nash equilibrium as a standard GAN, but through a different and more generalized class of loss functionals, such as hinge loss. This experimental design selection is based on our attempt in get the pair of models to converge [14] and to exhibit more stable behavior than regular GANs during training [15].

Our main contributions are summarized as follows:

- We investigate whether GANs can be used to learn representations of natural language in an unsupervised setting at the document, sentence, and aspect level.
- Among the various methods of learning representations, we focus on deep learning methods to yield more abstract—and, ultimately, more useful—representations.
- We bridge the unsupervised learning approach with GANs and denoising autoencoders.
- We revisit the traditional GAN framework from an alternative energy-based perspective.
- We propose a neural network architecture that is based on a variation of the energy-based GAN formulation [10] for generative adversarial training. Our contribution is based on the use of a simple hinge loss, at the point when the system reaches convergence, so that the generator of the energy-based GAN produces points that follow the underlying data distribution.
- We propose to use an autoencoder architecture as a discriminator in which the energy is a reconstruction error.
- We focus on the unsupervised benefit of GANs to process a large amount of unlabeled data and not on its ability to generate new data.
- We conducted extensive experiments by leveraging data of different in types, lengths, and genres: the 20 Newsgroups corpus, the Movie Review (MR) Dataset, and the Finegrained Sentiment Dataset (FSD).

The rest of this paper is organized as follows. Section 2 discusses previous work on ML approaches that have been used in text analysis, with emphasis on deep neural network approaches at the document, sentence, and aspect level. Section 3 sets the problem on the scene, providing the motivation and the details of our proposed adversarial neural network architecture. Section 4 presents the experimental validation, including the datasets, the implementation we followed, and the experimental results. Section 5 discusses the analysis of the main findings, suggesting future research directions, and Section 6 finalizes the study by drawing a couple of meaningful conclusions.

2. Related Work

ML for NLP can be performed with a wide range of text formats, from multi-sentence reviews and comments, to single-word expressions of opinion. The most frequent approach is document-level classification. Another approach, sentence-level classification, limits the analysis to single sentences instead of whole documents. It is typically a harder problem to solve since there is not much information that can be used by the classifier, with sentences being usually much shorter than documents. Compared with document- and sentence-level analysis, aspect-level analysis is more Finegrained. Its task is to extract and summarize people's opinions expressed on the aspects of entities, which are also called targets.

2.1. Machine Learning Approaches for NLP Tasks

Document- or sentence-level classification is often implemented by using **supervised ML methods**. This involves the training of a model using a large body of annotated data which is topic-specific. Any existing supervised ML method can be applied to document-level classification, such as support vector machines (SVMs) or hidden Markov models. Particularly, Pang et al. [16] compared many ML methods on a movie review classifier, concluding that SVMs and Naive Bayes had the best performance overall. Document- or sentence-level regression has also received much attention since many problems cannot be solved with a positive–negative classification, which is frequently used for product reviews, where a 1–5-star rating is prevalent. Pang et al. [17] compared various regression methods, such as SVM regression, SVM multiclass classification, and one-versus-all. Qu et al. [18] extended the bag-of-words representation by exploiting negation and sentiment modifiers, which are more influential in regression than in classification problems. Mejova et al. [19] reviewed feature selection strategies, such as stemming, term frequency, n-grams, point-of-speech, and negation-enriched features. They concluded that a smaller set of features outperformed a larger set for big datasets.

On the other hand, **unsupervised ML** for NLP relies on the dominating influence of sentiment words and phrases to perform the classification without the use of costly annotated data. This has been achieved either by extracting the syntactic patterns of the sentences or by using sentiment lexicons. Both approaches rely on the measurement of the sentiment orientation (SO) of phrases and eventually of the whole document. Lexicon-based analysis calculates the SO values of words and phrases, summing up to the polarity of the whole document. Such classifiers can incorporate negation and intensification since such operations can be easily identified by lexicons. In general, lexicon-based sentiment classifiers show a positive bias, which can be fixed by adjusting the value of the rarer negative expressions [20]. One of their deficits is that lexicon-based methods do not perform well on domain-dependent data, making them less efficient when used for domains that are more challenging for sentiment analysis (SA), such as politics. This issue was partially addressed in [21], but supervised methods still outperform lexicon-based methods for domain-specific problems.

Recently, neural networks (NNs) have started expanding to the field of NLP in the form of both supervised and unsupervised representation learning methods. In terms of **unsupervised representation learning** [22], much of the early research into modern deep learning was developed and validated via this approach [23–26]. Unsupervised learning is promising due to its ability to scale beyond only the subsets and domains of data that can be cleaned and labeled given resources, privacy, or other constraints. This advantage is also its difficulty. While supervised approaches have clear objectives that can be directly optimized, unsupervised approaches rely on proxy tasks, such as reconstruction, density estimation, or generation, which do not directly encourage useful representations for specific tasks. As a result, much work has gone into designing objectives, priors, and architectures meant to encourage the learning of useful representations.

Despite these difficulties, there are notable applications of unsupervised learning. Pretrained word vectors are a vital part of many modern NLP systems [27]. These representations, learned by modeling word co-occurrences, increase the data efficiency and generalization capability of NLP systems [28,29]. How to learn representations of phrases, sentences, and documents is an open area of research. Inspired by the success of word vectors, Kiros et al. [30] proposed skip-thought vectors, a method of training a sentence encoder by predicting the preceding and the following sentence. The representation learned by this objective performs competitively on a broad suite of evaluated tasks. More advanced training techniques, such as layer normalization [31], further improved results. However, skip-thought vectors are still outperformed by supervised models, which directly optimize the desired performance metric on a specific dataset. This is the case for both text classification tasks, which measure whether a specific concept is well encoded in a representation, and more general semantic similarity tasks. This occurs even when the datasets are relatively small by modern standards, often consisting of only a few thousand labeled examples.

2.2. Deep Neural Network Approaches at Document, Sentence, and Aspect Level

In this subsection, we review the latest research efforts, presented in Table 1, with respect to the application of NNs for NLP tasks, focusing on document-, sentence-, and aspect-level analysis. We would like to clarify at this point that we split Table 1 into three subsections to make a clear distinction among the works conducted with respect to the three aforementioned levels of analysis.

Table 1. Main characteristics of some of the latest published deep neural network models for sentiment analysis (SA) tasks. The table reports the level of analysis, the type of the deep neural network that has been applied (model), the datasets used, the type of task, and the evaluation metrics. The latter are labeled as follows: ccuracy (Acc.), Macro-F1 measure, Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Pearson correlation coefficient (r).

Work	Level of Analysis	Model	Dataset	Task	Evaluation Metrics
Rahman et al. [32]	Docs.	LSTM	Large Movie Review (50,000 reviews)	classification	Acc.: 80%

Lai et al. [33]	Docs.	RCNN	20 Newsgroups;
			Fudan Set;
			ACL Anthology Network;
			SST
				classification	Macro-F1: 96.49%
					Acc.: 95.2%
					Acc.: 49.19%
					Acc.: 47.21%

Shen et al. [34]	Docs.	CNN + BLSTM	Large Movie Review (50,000 reviews)	classification	Acc.: 89.7%
Yender and Verna [35]	Docs.	CNN + LSTM	Large Movie Review (50,000 reviews)	classification	Acc.: 89.5%

Liu et al. [36]	Docs.
	Sents.
		RNN	SST1 (Sents.);
			SST2 (Sents.);
			Movie Reviews (subj/obj. reviews);
			Large Movie Review (50,000 reviews)
				classification	Acc.: 49.6%
					Acc.: 87.9%
					Acc.: 94.1%
					Acc.: 91.3%

Chen et al. [37]	Sents.	BiLSTM-CRF
		CNN
			MPQA opinion corpus;
			SST;
			Movie Reviews (polarity v1.0)
				classification; target extraction	-
					Acc.: up to 88.3%
					Acc.: 82.3%

Wang X et al. [38]	Sents.	CNN + RNN	SST1;
			SST2;
			Movie Reviews
				classification	Acc.: 51.50%
					Acc.: 89.95%
					Acc.: 82.28%

Conneau et al. [39]	Sents.	Very Deep CNN	Product Reviews; News	classification	not reported
Wang J et al. [40]	Sents.	CNN + LSTM	SST; Chinese VA Texts	dimensional regression	RMSE/MAE/r: 1.341/0.987/0.778 RMSE/MAE/r: 0.874/0.689/0.557
Du et al. [41]	Docs.	Deep CNN	Amazon reviews	aspect classification	Acc.: 94.38%
Wang Y et al. [42]	Sents.	attention based LSTM	SemEval 2014 (Task 4)	aspect (binary) classification	Acc.: 89.9%
Poria et al. [43]	Sents.	Deep CNN	SemEval 2014; Aspect-based dataset	aspect extraction; classification	Acc.: up to 87.2%
Tang et al. [44]	Sents.	Deep Memory Network	SemEval 2014	aspect classification	Acc.: 80.95%

Rahman et al. [32] introduced a variation of long short-term memory (LSTM) and tested it on the Large Movie Review Dataset of 50K **documents**. It showed that even though accuracy was not increased compared to normal LSTM models, the stability and consistency were improved. Moreover, [33] introduced a recurrent convolutional neural network (RCNN) which applied a bidirectional recurrent structure to capture the contextual information of the document and then employed a max-pooling layer to capture the key components of the text. The researchers in [34] performed comparison tests with single CNN, single LSTM, and a combination of CNN and LSTM on the Large Movie Review Dataset for sentiment classification, showing that the CNN + LSTM network performed better than the respective standalone NNs. A similar research effort was conducted by [35] by combining CNN and LSTM to produce multiple variations which achieved state-of-art performance. Once again, the Large Movie Review Dataset was used, but the authors expressed their belief that the proposed model has potential in both audio and video. Finally, Liu et al. [36] experimented with RNNs, introducing the concept of multitask learning, which connects all the related tasks into a single system, trained jointly. They performed both document-level classification on the Large Movie Review Dataset and sentence-level classification on the Stanford Sentiment Treebank (SST), achieving state-of-art performance results after fine-tuning.

Moreover, NNs have been widely used for **sentence-level** classification as well. Particularly, Chen et al. [37] proposed a CNN–LSTM model which first classifies the sentences into non-target, one-target, and multi-target sentences. Their model was tested on many datasets (Stanford Sentiment Treebank, Movie Reviews, and Amazon product reviews), and it achieved state-of-the-art performance on some of them. Based on the fact that the combination of CNN and RNN is considered a very popular approach, the authors of [38] proposed their variant, which achieved high accuracy on the typical Stanford Sentiment Treebank and Movie Reviews datasets. Moreover, Conneau et al. [39] used a very deep convolutional network that consisted of 29 layers, resulting in improved performance and proof of the "benefit of depth" for NLP tasks as well. To be more precise, the authors evaluated their model on eight different datasets comprising Movie Reviews and news, testing various NLP tasks such as SA, news categorization, and topic classification. Moreover, Wang J et al. [40] proposed a regional CNN–LSTM model which used an individual sentence as the region for the extraction of affective information. The model was trained with the Stanford Sentiment Treebank dataset and Chinese Valence Arousal (VA) texts using 2K sentences from social forums, and it performed better than CNNs, RNNs, or LSTMs for valence and arousal prediction, respectively.

Finally, **aspect-level analysis** has been also explored. Du et al. [41] modeled both sentiment and syntactic context under specific aspects to acquire better word embeddings, which were given as input to a CNN for sentiment classification of Amazon product reviews. Their results showed an improvement compared to traditional word-embedding methods. Wang et al. [42] applied LSTMs for aspect-based sentiment classification, achieving a state-of-the-art performance of 89.9%. The model was evaluated on the SemEval 2014 dataset while the word embeddings were initialized by Glove (https://nlp.stanford.edu/projects/glove/), capturing the important parts of a sentence when different aspects were given. Poria et al. [43] tested a deep CNN for aspect extraction using seven levels of NNs. The SemEval 2014 dataset and an aspect-based SA dataset were used for the evaluation and showed improvement in precision and recall. Deep memory networks have also been applied to aspect-level text classification and have shown comparable results to LSTMs while being 15 times faster at the same time [44].

After summarizing the current and previous research efforts (in Table 1) that have been conducted within the area of deep neural networks at the document, sentence, and aspect levels of analysis, we observed that, in most cases, networks such as CNNs or recurrent neural networks (RNNs) and particular bidirectional long short-term memory (BLSTMs) networks have been applied; thus, very little work has been conducted on text analysis using generative models and particular GANs. Considering that one of our paper's aims is to examine whether GANs can be used to learn representations of NL texts in an unsupervised manner, in the following Section 2.3, we summarize most of the state-of-the-art research works that have used the GAN architecture. Additionally, motivated by the fact that the GAN neural network structure can integrate various loss functions, our proposed model was designed to have a better degree of freedom. Moreover, it is expected to provide promising solutions for creatively producing data that are meaningful to humans.

2.3. Generative Adversarial Networks for NLP Tasks

In this section, we review recent research on discovering rich structure in natural language with variational autoencoders (VAEs) [3] and GANs [4]. Evaluating deep generative models has been challenging so far. To the best of our knowledge, there are very few works on text analysis using GANs. Zhang et al. [45] proposed a framework for employing LSTM and CNN for adversarial training to generate realistic text. The latent code z was fed to the LSTM generator at every time step, while CNN acted as binary sentence classifier which discriminated between real data and generated samples.

RNN extensions, such as LSTMs or gated recurrent units (GRUs) that retain long-term memory of tokens, have been shown to work well in practice using maximum likelihood estimation.

However, training using maximum likelihood has its downsides, like exposure bias. This refers to the situation in which, during training, the prediction of the next word conditioned on the previous word becomes infeasible since the previous word may not have been seen in the training data. If the generator makes an error early on in the generation process, the generated sentence will keep diverging further away as more words are generated. To solve this issue, a few things have been tried in the past, like scheduled sampling [46] and having task-specific sequence scores. One problem with applying GAN to text is that the gradients from the discriminator cannot properly back-propagate through discrete variables. In [45], this problem was solved by making the word prediction at every time "soft" at the word-embedding space. Moreover, Yu et al. [47] proposed bypassing this problem by modeling the generator as a stochastic policy. The reward signal came from the GAN discriminator, judged on a complete sequence, and was passed back to the intermediate state-action steps using Monte-Carlo search. For text, it is possible to create oracle training data from a fixed set of grammars and then evaluate generative models based on whether (or how well) the generated samples agree with the predefined grammar [48]. Considering that it is hard to make a good evaluation for generating text since there is no objective way to assess whether an artificial sentence is more plausible or realistic than another, some works have used BLEU scores of samples on a large amount of unseen test data. The ability to generate similar sentences to unseen real data is considered a measurement of quality [47].

Notably, in terms of sequential data generation with GANs, an alternative approach based on reinforcement learning was used to train the GAN. We are aware of only one preliminary work using GANs to generate continuous-valued sequences, and it aimed to produce polyphonic music using a GAN with an LSTM generator and discriminator [49]. A work with respect to controllable text generation [50] applied the variable autoencoder (VAE) together with controllable information to generate category sentences. Finally, Zhang et al. [45] and Semeniuta et al. [51] used GANs for text generation and achieved state-of-the-art results. Finally, another category of approach is the conditional GANs [52] that condition the model on additional information and, therefore, allow us to direct the data generation process. This approach has been mainly used for image generation tasks [5,53]. Recently, conditional GAN architectures have been also used in NLP, including translation [47] and dialogue generation [54].

After summarizing the work conducted in the area of GANs for NLP tasks, we can observe that even though the research effort has been shifted to GAN approaches for NLP tasks, whose discriminator and generator models are mainly CNNs and/or RNNs (BLSTMs), still a number of limitations and development trends remain unsolved.

With this in mind, our attempt in this paper, presented in the following Section 3, is to investigate whether GANs are a suitable model selection to learn representations of natural language in an unsupervised setting at the document, sentence, and aspect level. Additionally, we revisit the traditional GAN framework from an alternative energy-based perspective. Particularly, we propose a neural network architecture that is based on a variation of the energy-based GAN formulation [10] for generative adversarial training. Our contribution is based on the use of a simple hinge loss, at the point when the system reaches convergence, so that the generator of the energy-based GAN produces points that follow the underlying data distribution. We propose to use a denoising autoencoder architecture as a discriminator in which the energy is a reconstruction error. Our experimental design selection is based on our attempt to get the pair of models to converge [14] and to exhibit more stable behavior than regular GANs during training [15].

3. Proposed Approach

In this section, we attempt to set the scene of our proposed approach by highlighting the limitations of the current energy-based models and discuss the advantages of GANs. GANs have great significance to the development of generative models. As a powerful class of generative methods, GANs solve the problem of generating data that can be naturally interpreted. Especially for the generation of high-dimensional data, the adopted neural network structure does not limit the generation dimension, which greatly broadens the scope of the generated samples. Moreover, the neural network structure can integrate various loss functions, thereby increasing the degree of freedom of the model design. As far as the energy-based models are concerned, they have been used to capture dependencies over variables by defining an energy function. The energy function associates each configuration of the variables with a scalar energy value. Lower energy values should be assigned to more likely or plausible configurations and, conversely, higher energy values should go to others. This has been used, for example, to estimate the probability distribution based on a Boltzmann distribution defined by an energy function and an appropriate normalization factor. In this case, the energy function is defined to assign a probability value that is not normalized. The normalization factor plays an important role by constraining the energy function to properly estimate the probability distribution. However, it introduces difficulties during the learning procedure, which requires an appropriate number of samples and makes the learning progress slow and noisy or requires certain model structures to get the samples. To overcome these limitations, we present in the following subsection our proposed neural network architecture.

3.1. Problem Formulation

We propose a model whose discriminator is viewed as an energy function that attributes low energies to the regions near the data manifold and higher energies to other regions, while its generator is seen as being trained to produce contrastive samples with minimal energies. With the term contrastive samples, we refer to a data point that causes an energy pull-up, such as an incorrect label in supervised learning and points from low data density regions in unsupervised learning. Our aim is to train the discriminator to assign high energies to these generated samples. Viewing the discriminator as an energy function, our proposed system allows us to use a wide variety of architectures and loss functionals—in addition to the usual binary classifier with logistic output—that have been introduced in the recently proposed "Energy-Based Generative Adversarial Network" model (EBGAN) [10]. However, our proposed model is a modified version of the EBGAN framework since we propose the use of an autoencoder architecture, with the energy being the reconstruction error, in place of the discriminator. Particularly, we suggest that rather than using a single bit of target information to train the model, the reconstruction-based output offers diverse targets for the discriminator. With the binary logistic loss, only two targets are possible, so, within a mini-batch, the gradients corresponding to different samples are most likely far from orthogonal. This leads to inefficient training, and reducing the mini-batch sizes is often not an option on current hardware. On the other hand, the reconstruction loss will likely produce very different gradient directions within the mini-batch, allowing for a larger mini-batch size without a loss of efficiency. Moreover, we decided to use the autoencoders since they have traditionally been used to represent energy-based models and arise naturally. When trained with some regularization terms, autoencoders have the ability to learn an energy manifold without supervision or negative examples. This means that even when an energy-based autoencoding model is trained to reconstruct a real sample, the discriminator contributes to discovering the data manifold by itself. To the contrary, without the presence of negative examples from the generator, a discriminator trained with binary logistic loss becomes pointless.

Our proposed work is also inspired by the work of Kim and Bengio [13]. However, it differs in the following way. Their approach uses a standard probabilistic GAN cast into an energy model (using Gibbs distributions), allowing them to learn a discriminator that models the distribution when Nash equilibrium is reached, but it still has mixing problems, especially with deep models that slow the learning. On the contrary, our approach gets rid of the probabilistic setting, while presenting the same Nash equilibrium as a standard GAN, but through a different and more generalized class of loss functionals.

The computational procedure and structure of our proposed GAN model are depicted in Figure 1. We use differentiable functions D and G to represent the discriminator and the generator, respectively. Their inputs are real data x and random variables z, respectively. $G(z)$ represents the sample generated by G according to the distribution p_{data} of real data, and x' corresponds to the novel synthetic data samples. If the input of discriminator D is from the real data x, D should classify it to be true. If the input is from $G(z)$, D should classify it to be false. The purpose of D is to achieve the correct classification of the data source, while the purpose of G is to approximate the performance of the generated data $G(z)$ on D (i.e., $D(G(z))$) with the performance of real data x on D (i.e., $D(x)$). The adversarial optimization process improves the performance of D and G gradually. Eventually, when the discrimination ability of D has been improved to a high level but cannot discriminate the data source correctly, it is thought that the generator G has captured the distribution of real data.

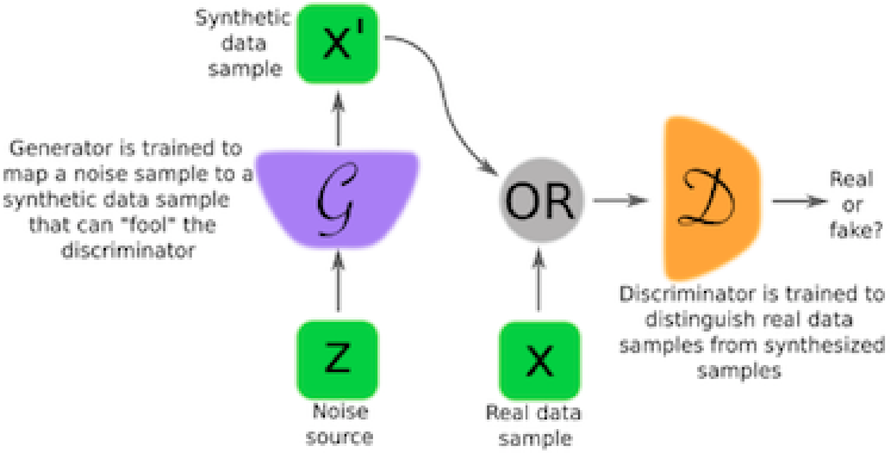

Synthetic data sample
X'
Generator is trained to map a noise sample to a synthetic data sample that can "fool" the discriminator
G
OR
D
Real or fake?
Discriminator is trained to distinguish real data samples from synthesized samples
Z
Noise source
X
Real data sample

Figure 1. The architecture of a generative adversarial network (GAN) model, consisting of a discriminator D and a generator G model which are learned during the training process of a GAN.

3.2. Adversarial Document-Level Neural Network Architecture

The original GAN formulation [4], presented in Figure 1, consists of a min-max adversarial game between a generative model G and a discriminative model D. $G(z)$ is a neural network that is trained to map samples z from a prior noise distribution $p(z)$ to the data space. $D(x)$ is another neural network that takes a data sample x as input and outputs a single scalar value representing the probability that x came from the data distribution instead of $G(z)$. D is trained to maximize the probability of assigning the correct label to the input x, while G is trained to maximally confuse D, using the gradient of $D(x)$ with respect to x to update its parameters. This min-max game can be optimized by the following risk, given by Equation (1), and is typically implemented with neural network models; however, these models could be implemented by any form of differentiable system that maps data from one space to another.

$$\phi = min_G max_D E_{x \sim p(data)}[logD(x)] + E_{z \sim p(z)}[log(1 - D(G(z)))] \tag{1}$$

One shortcoming of this model is that there are no explicit means for inference, and so it is unclear how GANs could be applied to unsupervised representation learning. In [4], two possible solutions were suggested and have been explored by the research community in subsequent works. The first approach is to train another network to do inference, learning a mapping from x back to z [55], with a variation of this method being to instead use the adversarial training process to regularize an autoencoder's representation layer [56]. The second idea is to use internal components of the discriminator network as a representation [5]. Nevertheless, both approaches fail to result in an architecture with stable training across a range of datasets and model hyperparameters when using a probabilistic discriminator network. Thus, our approach to improving GAN training is to assess the empirical symptoms that are experienced during training [5] by switching to use the energy-based GAN architecture, where the discriminator is an autoencoder [10]. Document representations can then be formed from the encoded representation of the discriminator.

Particularly, our proposed model's architecture is formulated as follows: Let $x_i \in \{0,1\}^V$ be the binary bag-of-words representations of a document, where V is the vocabulary size and x_i is a binary value indicating whether the ith word is present in the document or not. We define a feedforward generator network $G(z)$ that takes a vector $z \in R^{h_g}$ as input and produces a vector $\hat{x} \in R^V$, with h_g being the number of dimensions in the input noise vector (sampled from $N(0,1)$). We also define a discriminator network $D(x)$, seen as an energy function, that takes vectors $x \in R^V$ and produces an energy estimate $E \in R$.

One main difference compared to the work of [10] is that we used a denoising autoencoder (DAE) as our energy function, as the DAE has been found to produce superior representations to the standard autoencoder [57]. In this work, we used single encoding and decoding layers, so the encoding process is given in Equation (2):

$$h = f(W^e x^c + b_e) \tag{2}$$

where W^e is a set of learned parameters referring to a deterministic mapping from a data space into the latent/representation space, b_e is a learned bias term, f is a nonlinearity, x^c is a corrupted version of x, and $h \in R^{h_d}$ is the hidden representation of size h_d, with h_d being the size of the denoising autoencoder (DAE) hidden state. The decoding process is then given by:

$$y = W^d h + b_d \tag{3}$$

where W^d and b_d are another learned set of weights and bias terms. The final energy value is the mean squared reconstruction error:

$$E = \frac{1}{V}\sum_{i=1}^{V}(x_i - y_i)^2 \tag{4}$$

Our proposed GAN model's architecture is presented in Figure 2. The energy function is trained to push down on the energy of the real samples x, and to push up on the energy of the generated samples $\hat{x}$ [10]. This is given by Equation (5), where f_D is the value to be minimized at each iteration and m is a margin between positive and negative energies. In other words, the energy function outputs low values on the data manifold and higher values everywhere else.

$$f_D(x,z) = D(x) + max(0, m - D(G(z))) \tag{5}$$

At each iteration, the generator G is trained adversarially against D to minimize f_G. In other words, the generator G learns to pick points where the energy should be increased, while the discriminator D is viewed as a learned objective function. Aligned with the generator's role, the latter model is trained to create samples that will fool the discriminator, so that the adversarial game that is played between the two networks will converge on a saddle point that is a local minimum for the discriminator and a local maximum for the generator.

$$f_G(z) = D(G(z)) \tag{6}$$

In a similar way, sentence and aspect representations can be formed from the encoded representation of the discriminator, and their network's structure is built in a manner similar to the adversarial document's structure. In particular, the energy function for sentence and aspect representations are trained to push down the energy of real samples x_{sent} and x_{asp}, and to push up on the energy of generated samples $\hat{x}_{sent}$ and $\hat{x}_{asp}$, respectively [10]. This is given by Equations (7) and (8), respectively, where $f_{D_{sent}}$ and $f_{D_{asp}}$ are the values to be minimized at each iteration and m is a margin between positive and negative energies. Thus, the energy function outputs low values on the data manifold and higher values everywhere else.

$$f_{D_{sent}}(x_{sent}, z_{sent}) = D(x_{sent}) + max(0, m - D_{sent}(G(z)_{sent})) \tag{7}$$

$$f_{D_{asp}}(x_{asp}, z_{asp}) = D(x_{asp}) + max(0, m - D_{asp}(G(z)_{asp})) \tag{8}$$

Finally, as we explain above, at each iteration, regarding the adversarial min-max game between the generator and the discriminator for the sentence and the aspect representations, the generators G_{sent} and G_{asp} are trained adversarially against D_{sent} and D_{asp} to minimize $f_{G_{sent}}$ and $f_{G_{asp}}$, respectively. In other words, the generators G_{sent} and G_{asp} learn to pick points where the energy should be increased, while the discriminators D_{sent} and D_{asp} are viewed as learned objective functions, depicted in Equations (9) and (10), respectively.

$$f_{G_{sent}(z_{sent})} = D_{sent}(G_{sent}(z_{sent})) \tag{9}$$

$$f_{G_{asp}(z_{asp})} = D_{asp}(G_{asp}(z_{asp})) \tag{10}$$

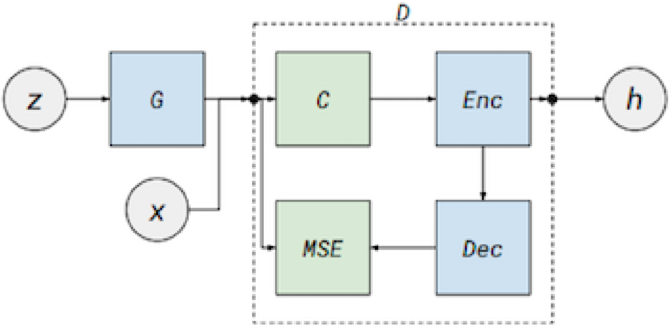


Figure 2. Architecture of the proposed GAN for learning representation at the document, sentence, and aspect level. The network has the following parts: generator *G*, denoising autoencoder (DAE) encoder *Enc*, and decoder *Dec*, a corruption process *C*, and a discriminator *D*.

4. Experimental Validation

This section describes the document-, sentence-, and aspect-level datasets (Section 4.1), the implementation design decisions and the parameters' actual selection of our proposed GAN architecture model (Section 4.3) used in the experimental evaluation, and, finally, the experimental results (Section 4.4).

4.1. Datasets

Selecting appropriate datasets is important for the evaluation of NLP models. There are many publicly available datasets that have been used extensively in academia and come from the SA subfield of NLP; the use of such datasets is fueled by an unprecedented flood of social network activity over the last decade and an interest in processing the social media enhanced with sentiment. Among them, the most popular are those datasets provided by the Stanford University, the SST1 and SST2, the Large Movie Review Dataset, the MPQA opinion corpus [37], a dataset of Amazon reviews [41], the ACL Anthology, and the 20 Newsgroups [33]. Below, we give a detailed description of the three benchmarks we used to perform document, sentence and aspect analysis.

(1) The 20 Newsgroups dataset (https://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups): The 20 Newsgroups dataset is a collection of approximately 20,000 newsgroup documents. It is considered a popular dataset for experiments in text applications of ML techniques, such as text classification and text clustering. The data are organized into 20 different newsgroups, each

corresponding to a different topic. Some of the newsgroups are very closely related to each other (e.g., comp.sys.ibm.pc.hardware/comp.sys.mac.hardware), while others are highly unrelated (e.g., misc.forsale/soc.religion.christian). The split between the training (50%), the validation (10%), and the test (40%) set is based upon messages posted before and after a specific date, while cross-posts (duplicates) and newsgroup-identifying headers (Xref, Newsgroups, Path, Followup-To, Date) are excluded. The 20news-bydate.tar.gz version of the 20 Newsgroups dataset consists of 18,786 documents (postings); however, after additional preprocessing (i.e., excluding a number of duplicates and removing some headers) the final number of documents is 18.821. We further split the data into 10.163 for training, 1.130 as the validation set, and 7.528 test documents.

(2) Movie Review Dataset (subjectivity dataset v1.0) (https://www.cs.cornell.edu/people/pabo/movie-review-data/): This dataset includes 5000 subjective and 5000 objective processed sentences. With respect to the level of analysis, each line in these two files (subjective and objective) corresponds to a single sentence or snippet; all sentences (or snippets) are down-cased. Only sentences or snippets containing at least 10 tokens were included. The sentences and snippets are labeled automatically. From our point of view, it seems adequate to use the Movie Review Dataset provided by Pang and Lee that is freely available. The fact that many articles in SA discuss this dataset and have used it to validate their own methods and approaches makes it an ideal candidate from the benchmarking angle. Finally, for our experimental setting, we randomly split each collection into a training and a test set of 75% and 25%, respectively. In this dataset, the validation set refers to 10% of the training set. We decided to split the training data into initial training data and a validation test set to further avoid gradual overfitting on the test data and to avoid getting unrealistically good results on our final test set.

(3) Finegrained Sentiment Dataset (https://github.com/oscartackstrom/sentence-sentiment-data): The Finegrained Sentiment Dataset contains 294 product reviews from various online sources that are manually annotated with sentence-level sentiment. The data are approximately balanced with respect to the domain (books, DVDs, electronics, music, video games) and overall review sentiment (positive, negative, neutral), and the sentiment labels are assigned to sentences by two annotators. With respect to the Finegrained Dataset's product reviews, the sentiment labels refer to positive and negative opinions on the different aspects of the product, although the general sentiment on the product could be positive or negative. The FSD collection includes a total of 2243 polar sentences: 923 positive sentences and 1320 negative sentences. Once again, we randomly split each collection into a training and a test set of 75% and 25%, respectively. In this dataset, the validation set refers to 10% of the training set. The experimental setup for the three dataset collections is reported in Table 2.

Table 2. Test collections for experiments on the use of generative adversarial networks to learn distributed representations of natural language at the document, sentence, and aspect level.

Datasets	Training	Validation	Test
20 Newsgroup (http://qwone.com/~jason/20Newsgroups/)	10.163	1130	7528
Movie Reviews [58]	7424	76	2500
Finegrained Dataset (FSD) [59]	2582	287	956

4.2. Baseline System

We established a simple GAN model consisting of a three-layer feedforward generator and discriminator networks as our baseline. The parameters of the networks were optimized using the development set. The baseline was implemented with similar architecture, serving as a fair comparison with the proposed method (e.g., number of layers). The key difference with our proposed model is that the proposed model's discriminator is a denoising autoencoder.

4.3. Implementation

We further preprocessed all three datasets in order to clean them of any noisy data to further reduce the complexity of our datasets. We decided to remove irrelevant features, such as common stop words (such as I or and); we tokenized the datasets to split up their words into terms of tokens; and we stemmed the latter to reduce the tokens into a single type, normally a root word. As such, the stemming process reduced redundant words in our datasets. Our aim is to create a vocabulary of approximately 40,000 different word stems. From these, approximately 2000 of the most popular ones were kept in the training datasets. As a result, each posting is represented as a vector containing a 2000 word count.

Even though word embedding is currently the-state-of-art of the NLP field in resolving text-related problems, due to a number of limitations that presented, we decided to use bag-of-words (BoW). Particularly, one limitation of individual word embeddings is their inability to represent phrases where the combination of two or more words does not represent the combination of the meanings of the individual words [60]. Another limitation comes from learning embeddings based only a small window of surrounding words: sometimes words, such as *good* and *bad*, share almost the same embedding [61], which is problematic if used in tasks such as SA [62]. Moreover, a general caveat for word embeddings is that they are highly dependent on the applications in which they are used [63]; as such, BoW works better than the current word-embedding models (Word2Vec, GloVe) for our examined scenario.

We trained our networks using Keras [64] with Tensorflow as the back-end [65]. Following [66], to make a direct comparison, we set our representation size h_d (the size of the denoising autoencoder (DAE) hidden state) to 50. The generator input noise vector h_g was also set to be the same size. The generator is a three-layer feedforward network, with rectified linear unit (ReLU) activations in the first two layers, and a sigmoid nonlinearity in the output layer. Layers 1 and 2 are both of size 300, with the final layer being the same size as the vocabulary. Layers 1 and 2 use batch normalization [15]. We decided to use the latter, as it has been recommended for use in both networks in order to stabilize training in deeper models. The discriminator encoder consists of a single linear layer followed by a leaky ReLU nonlinearity function (with a leak of 0.02). We decided to do so, as it has been shown that using a leaky ReLU activation functions between the intermediate layers of the discriminator, giving a superior performance to that when using regular ReLUs [5]. The decoder is a linear transformation back to the vocabulary size. We optimized both G and D using the first-order gradient-based Adam optimizer [67] with an initial learning rate of 0.0001. Our denoising autoencoder (DAE) corruption process was to randomly set to zero 40% of the input values, and we used a margin size m of 5% of the vocabulary size.

We followed the same validation procedure as Salakhutdinov and Hinton [66], and we set the validation set to perform model selection of other hyperparameters, such as the learning rate and the number of learning passes over the training set (based on early stopping). We also tested the use of a hidden layer hyperbolic tangent nonlinearity, given in Equation (11), instead of the sigmoid, and always used the best option based on the validation set performance. Finally, our proposed GAN model was trained with a learning rate of 0.01 and using the tanh activation function.

$$tanh(x) = (\exp(x) - \exp(-x))/(\exp(x) + \exp(-x)) \tag{11}$$

4.4. Experimental Results

In order to conduct a comprehensive evaluation of our proposed method, we used three types of datasets from different domains. It is also important to note that the evaluated datasets, as described in Section 4.1, vary greatly in length of the analyzed preprocessed text, ranging from 18.821 and 10.000 to 3.825 documents and sentences, respectively. This diversity enabled us to evaluate the robustness of our proposed approach in multiple experimental settings. As of yet, there is no consensus regarding the best evaluation metric with respect to the GAN's performance [68].

Different metrics assess, both qualitatively and quantitatively, various aspects of the generation process, and it is unlikely that a single metric can cover all aspects. Currently, two widely accepted scores, Inception Score [15] and Frechet Inception Distance, rely on pretrained deep networks to represent and statistically compare original and generated samples. However, these scores have been applied and tested only on the CV domain and not on SA tasks. As far as the SA tasks are concerned, accuracy is the most commonly used measure, taking into account the latest published works presented in Table 1, mainly when CNNs and/or RNNs are examined. Thus, we maintain that there is good reason to consider the other measures as well.

Taking into account that our proposed GAN architecture is based on a variation of the energy-based GAN [10] and to further make a direct comparison with the latter work, we decided to use the precision/recall curves measure to evaluate the performance of the proposed model used in our study. Precision is defined as the percentage of relevant items out of the "top X" items retrieved by our algorithm, while recall is the percentage of retrieved relevant items of all the relevant items in the dataset [69].

For completeness, we further decided to use the F-measure metric to evaluate our proposed system's performance. The F-measure enables factoring both precision and recall into a single value, thus representing their harmonic mean [70]. This measure is calculated using the following formula:

$$F - measure = \frac{(1+\beta^2) \times recall \times precision}{(\beta^2 \times precision) + recall} \tag{12}$$

In our experiments, we used β = 1, thus giving equal weight to the two measures.

Although accuracy is the most commonly used measure for SA tasks, an additional reason to consider the above-presented measures is the issue of dataset imbalance [71–73]. To be more precise, in many cases, SA datasets suffer from imbalances in class distribution, where the number of instances belonging to one class significantly outnumbers that which belongs to another. With respect to the datasets we used to evaluate our system, the 20 Newsgroup dataset and the Finegrained Sentiment Analysis datasets contain such significant imbalances in class distributions.

As such, the advantage of our selected evaluation measure is that it enables the evaluation of our performance on the top X instances that are relevant, sorted by their probability of belonging to a certain class of relevant instances. This measure is often used in information retrieval when measuring the performance of a query: as the user is not likely to review thousands of documents/sentences with respect to an aspect, the top ranking documents/sentences are more important and thus receive greater weight. This measure enables us to analyze the performance of the evaluated algorithm from multiple perspectives. Finally, it is also particularly useful for real-world applications, where often only the top X items are used in the evaluation process.

Table 3 presents the performance in terms of the precision, recall, and F-measure metrics for the models trained with a three-layer feedforward generator and the discriminator networks, consisting of a single linear layer (encoder) and a linear transformation back to the vocabulary size (decoder), respectively. We observe that the sentence-level Movie Reviews Dataset performs better, while the document-level 20 Newsgroups dataset provides slightly better precision results compared to the Finegrained sentences.

Table 3. Our proposed GAN model's performance as a generative model when using various amounts and types of data at the document, sentence, and aspect level, compared with a simple GAN, which serves as our baseline.

	Baseline			**Proposed GAN**		
Dataset	Precision	Recall	F-measure	Precision	Recall	F-measure
20 Newsgroups	0.2521	0.0001	0.996×10^{-4}	0.4188	0.0001	1.999×10^{-4}
	0.2099	0.0002	3.996×10^{-4}	0.4012	0.0002	3.996×10^{-4}
	0.1005	0.0005	9×10^{-4}	0.3648	0.0005	9.986×10^{-4}
Movie Reviews	0.3637	0.0001	1.999×10^{-4}	0.6376	0.0001	1.999×10^{-4}
	0.3637	0.0002	3.978×10^{-4}	0.6376	0.0002	3.998×10^{-4}
	0.3901	0.0005	9.871×10^{-4}	0.6202	0.0005	9.991×10^{-4}
Finegrained Dataset (FSD)	0.1022	0.0001	2.101×10^{-4}	0.3522	0.0001	2×10^{-4}
	0.1152	0.0002	3.993×10^{-4}	0.3483	0.0002	3.997×10^{-4}
	0.3185	0.0005	9×10^{-4}	0.3483	0.0005	9.985×10^{-4}

To visualize our produced results, we further created the precision–recall curves and we treated our model's performance as a retrieval task on the 20 Newsgroups Dataset, on the Movie Reviews Dataset, and on the FSD. The precision–recall curves for the recall values given in Table 3 are presented in Figure 3.

Figure 4 shows the visualizations of the learned representations created using the t-Distributed Stochastic Neighbor Embedding (t-SNE) toolkit [74]. We used this toolkit and created 2D projections of the data distribution. Particularly, Figure 4a–c show the distributions of the test document-, sentence-, and aspect-level test data learned by our proposed adversarial model, respectively. The documents belong to 20 different topics, the sentences of the Movie Reviews dataset belong to 2 different categories, while the sentences of the Finegrained dataset belong to 5 categories, with separate categories corresponding to different colored points in the subfigures. By using adversarial training, the feature distributions for the test samples from all three datasets are almost indistinguishable, demonstrating that our proposed approach is able to find common representation.

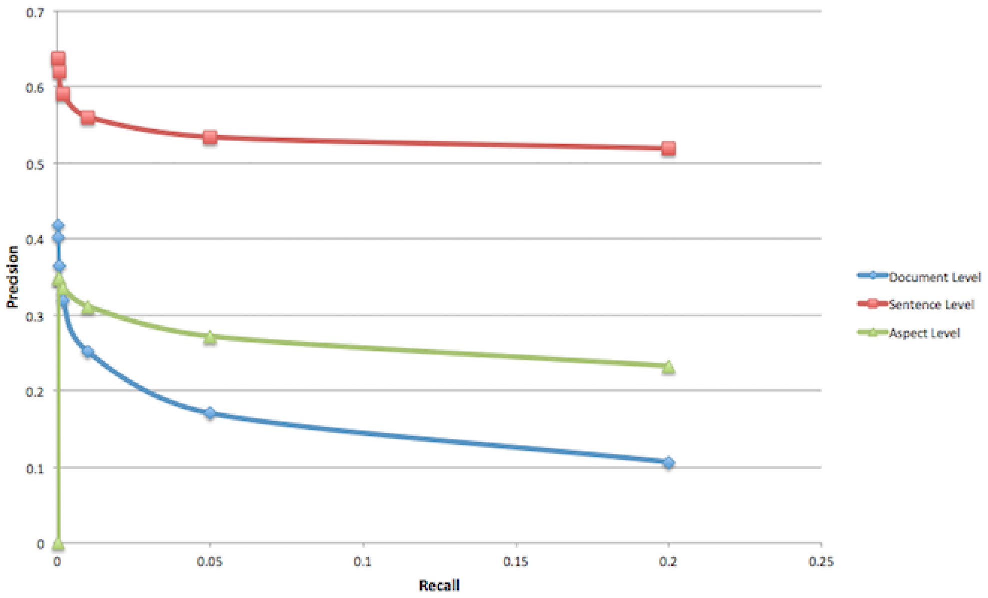
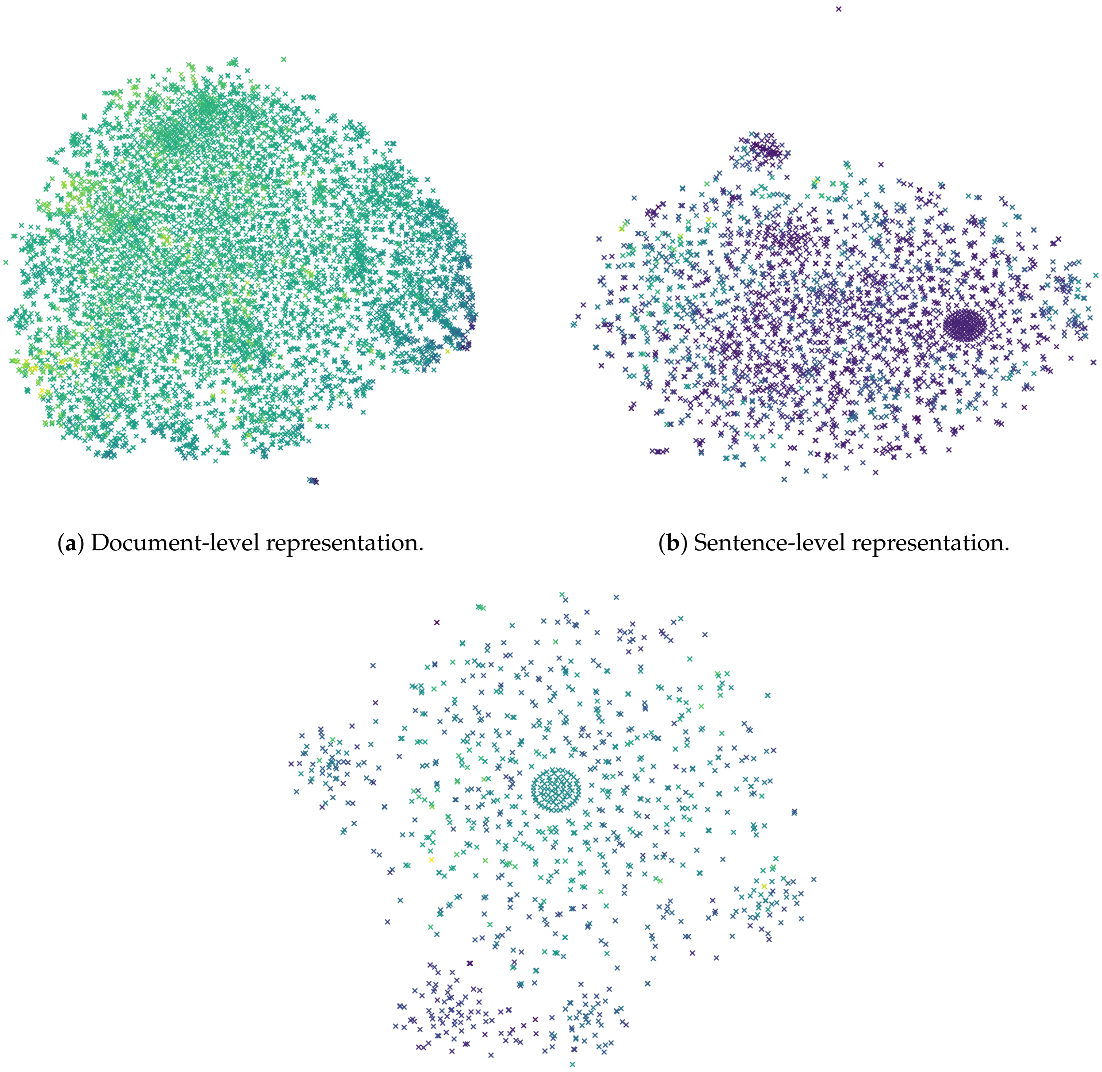


Figure 3. Precision–recall curves of our proposed GAN model on the 20 Newsgroups, the Movie Reviews, and the Finegrained datasets.



(**a**) Document-level representation.

(**b**) Sentence-level representation.

(**c**) Aspect-level representation.

Figure 4. t-Distributed Stochastic Neighbor Embedding (t-SNE) visualizations of the document, sentence, and aspect representations learned by our proposed adversarial model on the three test datasets. The color indicates different class labels. (**a**) Document-level representation using the GAN model; (**b**) Sentence-level representation using the GAN model; (**c**) Aspect-level representation using the GAN model.

5. Discussion

The current section discusses the results of the analysis carried out from an experimental design point of view. With respect to **the number of hidden layers** for feature representation, we first studied the effect of the number of hidden layers in both the generator and the discriminator. We varied the number of hidden layers, ranging from one to four first at the generator only, then at the discriminator, and, finally, at both networks simultaneously, and observed how the changes in feature representation affect the classification performance. This evaluation is conducted exclusively on the validation set of all three examined dataset collections. In most cases, two or three layers provide the best performance. Despite the theoretical existence of unique solutions, GAN training is challenging and often unstable for several reasons, as reported in [5,15,75]. One approach to improve GAN training is to assess the empirical "symptoms" that might be experienced during training. These symptoms include: difficulties in getting the pair of models to converge [5], the generative model collapsing to generate very similar samples for different inputs [15], and, finally, the discriminator loss converging quickly to zero [75], providing a non-reliable path for gradient updates to the generator. Based on our extensive experiments, we did not notice any significant performance increase due to the increase in the number of hidden layers in our proposed network (neither in the generator nor in the discriminator model for all three different types of datasets). Our initial aim was to select for each type of dataset the ideal "capacity" of hidden layers for both models of our network. However, our results are aligned with the work of [76], according to which even if providing successful solutions to the above-presented challenging issues when training GANs, building the GAN network still depends highly on the examined task/application.

To investigate **the performance results** achieved by our proposed model, we used all of the test data from the three datasets as queries, and we compared them to a fraction of the closest type of text of the examined dataset in the three training sets, from which we calculated the similarity based on the cosine similarity between the vector representations. The average number of returned documents and sentences having the same label as the query document (precision) were recorded. As far as the selected evaluation metric are concerned, we maintain that there is a good reason to consider precision–recall due to the issue of dataset imbalance. For information retrieval, precision is a measure of result relevancy, while recall is a measure of how many truly relevant results are returned. Thus, the precision–recall curves presented in Figure 3 show the trade-off between precision and recall for different thresholds. We observed that we achieved high scores for both precision and recall for the Movie Reviews Dataset (sentence level), suggesting that our proposed GAN architecture returns accurate results (high precision), as well as returning a majority of all positive results (high recall). Additionally, we observed that our aspect-level GAN system for the same recall value has 0.2% higher precision results compared to the document-level GAN system. Apart from that, our results suggest that considering that our proposed document-level GAN model achieves high precision but low recall, it returns very few results, but most of its generated documents are correct when compared to the training ones. Finally, after comparing our proposed model performance with the baseline performance, we observed that, in all cases, all three different benchmarks for the same recall values given in Table 3 outperform our GAN baseline. Overall, the most ideal system among the three, achieving both high precision and high recall while return many results generated correctly, is the sentence-level GAN model based on the Movie Reviews Dataset. Moreover, our selection of one evaluation metric to examine the performance of our GAN model is aligned with the work of [76], according to which attempting to evaluate GANs using different measures may lead to conflicting conclusions about the quality of the synthesized samples; the decision to select one measure over another depends on the application examined.

We also explored the feature representation when the GAN model is trained using the t-SNE toolkit. The objective of this evaluation is to visualize the distribution of the sparse matrices from the test data of the three datasets. This evaluation was implemented using the vector models using the 20 Newsgroups, the Movie Reviews, and the Finegrained corpora. For high-dimensional sparse data, it is helpful to first reduce the dimensions to 50 dimensions with truncated singular value decomposition (TruncatedSVD) and then perform t-SNE. This usually improves the visualization.

With respect to the hyperparameters' selection, it has been suggested by van der Maaten and Hinton [74] that perplexity values should range between 5 and 50. After experimenting in the range of 2–100, we observed that with too small perplexity values, local variations dominate, while with an image perplexity of 100, pitfalls are illustrated. Thus, we conclude that for our adversarial training to operate properly, the perplexity really should be smaller than the number of points. As such, for the needs of our experiments, we set the perplexity value to 40 instead of 30, the latter being the default value suggested. Moreover, each of the plots illustrated in Figure 4a–c was made with 1000 iterations with a learning rate (often called "epsilon") of 10, and reached a point of stability by step 1000. Keeping the perplexity value constant at 40, we experimented with the numbers of 10, 20, 60, 120, and 1000 steps, producing images for five different runs at a perplexity of 40. We observed that the first four were stopped before stability was reached, while the produced layouts seemed one-dimensional instead of two-dimensional with strange "pinched" shapes, indicating that the process was stopped too early. As a result, we conclude that there is no fixed number of steps that yields a stable result. In other words, different datasets can require different numbers of iterations to converge.

One challenging aspect in using the proposed approach is the difficulty of training adversarial networks. For example, Zhao et al. [10] noted that in NLP problems, the improvements of EBGANs were large for some types of noises, but less effective for others. They suggested that tuning the parameters could lead to better results. We also observed that the framework failed to converge for certain parameters, which is common in min-max problems. When properly trained, however, this powerful framework can satisfactory/elegantly solve one of the most important problems in NLP.

6. Conclusions

Adversarial networks have enjoyed much success in the CV domain [4,77], but to our best knowledge, they have not yet achieved comparably successful results when applied to SA tasks. This study proposes an elegant solution to the problem of learning representations at the document, sentence, and aspect level based on generative adversarial training in an unsupervised way. Particularly, this paper proposes a novel extension of generative adversarial networks that replaces the traditional binary classifier discriminator with one that assigns a scalar energy to each point in the generator's output domain. The discriminator minimizes a hinge loss, while the generator attempts to generate samples with low energy under the discriminator. We show that a Nash equilibrium under these conditions yields a generator that matches the data distribution (assuming infinite capacity). Experiments were conducted with the discriminator taking the form of an autoencoder, optionally including a regularizer that penalizes generated samples having a high cosine similarity to other samples in the mini-batch. Finally, we visualized the data representation of all three architectures by projecting the features into the layers of the proposed adversarial document-, sentence-, and aspect-level neural network architectures for SA, respectively.

There are several promising directions for future work highlighted by our proposed adversarial document-, sentence-, and aspect-level neural network architectures for SA. Considering that, in general, the neural network structure of our GAN models can integrate various loss functions, we do intend to increase the degree of freedom of the model design. Particularly, since our proposed approach is an alternative energy-based perspective of the GAN framework, it would be interesting to incorporate the family of energy-based loss functionals presented in [78] into the energy-based GAN models. Thus, the conditional setting presented in [79] is a promising setup to explore, and we expect that will attract more attention to a broader view of GANs from the energy-based perspective. Additionally, taking into account that GANs are also meaningful and instructive for semi-supervised learning, we do intend to use the training process of GANs to achieve pretrained data using unlabeled data. To be more precise, we aim first to use a large amount of unlabeled data to train our proposed GAN models, and, based on the understanding of the trained GANs over the unlabeled data, we intend then to use a small amount of labeled data to train the discriminative model for classification and regression tasks. Finally, our future plans include gaining an understanding of why the denoising autoencoder in the GAN discriminator appears to produce significantly better representations than a standard autoencoder, a result demonstrated by the experimental evaluation.

Author Contributions: A.V. conceived of the idea, designed and performed the experiments and analyzed the results, A.V. and G.C. drafted the initial manuscript and A.V., P.M. and A.S. revised the final manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252.
2. Rezende, D.J.; Mohamed, S.; Wierstra, D. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 1278–1286.
3. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *Mach. Learn.* **2013**, arXiv:1312.6114.
4. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Mach. Learn.* **2014**, 2672–2680, arXiv:1406.2661.
5. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *Mach. Learn.* **2015**, arXiv:1511.06434.
6. Gregor, K.; Danihelka, I.; Graves, A.; Rezende, D.; Wierstra, D. DRAW: A Recurrent Neural Network For Image Generation. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 1462–1471.
7. Bowman, S.R.; Vilnis, L.; Vinyals, O.; Dai, A.; Jozefowicz, R.; Bengio, S. Generating Sentences from a Continuous Space. In Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, Berlin, Germany, 11–12 August 2016; pp. 10–21.
8. Miao, Y.; Yu, L.; Blunsom, P. Neural variational inference for text processing. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; pp. 1727–1736.
9. Salant, S.W.; Switzer, S.; Reynolds, R.J. Losses from Horizontal Merger: The Effects of an Exogenous Change in Industry Structure on Cournot-Nash Equilibrium. *Q. J. Econ.* **1983**, *98*, 185–199.
10. Zhao, J.; Mathieu, M.; LeCun, Y. Energy-based generative adversarial network. *Mach. Learn.* **2016**, arXiv:1609.03126.
11. Deng, L. The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Process. Mag.* **2012**, *29*, 141–142.
12. Geman, S.; Geman, D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* **1984**, *6*, 721–741.
13. Kim, T.; Bengio, Y. Deep directed generative models with energy-based probability estimation. *Mach. Learn.* **2016**, arXiv:1606.03439
14. Yu, Y.; Gong, Z.; Zhong, P.; Shan, J. Unsupervised Representation Learning with Deep Convolutional Neural Network for Remote Sensing Images. In Proceedings of the International Conference on Image and Graphics, Shanghai, China, 13–15 September 2017; pp. 97–108.
15. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved techniques for training gans. In Proceedings of the 30th Conference on Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 2234–2242.
16. Pang, B.; Lee, L.; Vaithyanathan, S. Thumbs up: Sentiment classification using machine learning techniques. In Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing-Volume 10. Association for Computational Linguistics, Stroudsburg, PA, USA, 9–11 October 2002; pp. 79–86.
17. Pang, B.; Lee, L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Ann Arbor, MI, USA, 25–30 June 2005; pp. 115–124.
18. Qu, L.; Ifrim, G.; Weikum, G. The bag-of-opinions method for review rating prediction from sparse text patterns. In Proceedings of the 23rd International Conference on Computational Linguistics, Beijing, China, 23–27 August 2010; pp. 913–921.
19. Mejova, Y.; Srinivasan, P. Exploring Feature Definition and Selection for Sentiment Classifiers. In Proceedings of the Fifth International Conference on Weblogs and Social Media (ICWSM), Barcelona, Spain, 17–21 July 2011.
20. Taboada, M.; Brooke, J.; Tofiloski, M.; Voll, K.; Stede, M. Lexicon-based methods for sentiment analysis. *Comput. Ling.* **2011**, *37*, 267–307.
21. Zhang, L.; Liu, B. Extracting resource terms for sentiment analysis. In Proceedings of the 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand, 8–13 November 2011; pp. 1171–1179.

22. Olshausen, B.A.; Field, D.J. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vis. Res.* **1997**, *37*, 3311–3325.
23. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507.
24. Huang, F.J.; Boureau, Y.L.; LeCun, Y. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR'07Honolulu, HI, USA, 17–22 June 2007; pp. 1–8.
25. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.A. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 1096–1103.
26. Coates, A.; Ng, A.; Lee, H. An analysis of single-layer networks in unsupervised feature learning. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Lauderdale, FL, USA, 11–13 April 2011; pp. 215–223.
27. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
28. Pennington, J.; Socher, R.; Manning, C. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
29. Chen, D.; Manning, C. A fast and accurate dependency parser using neural networks. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 740–750.
30. Kiros, R.; Zhu, Y.; Salakhutdinov, R.R.; Zemel, R.; Urtasun, R.; Torralba, A.; Fidler, S. Skip-thought vectors. *Computa. Lang.* **2015**, 3294–3302, arXiv:1506.06726.
31. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *Mach. Learn.* **2016**, arXiv:1607.06450.
32. Rahman, L.; Mohammed, N.; Al Azad, A.K. A new LSTM model by introducing biological cell state. In Proceedings of the 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), Dhaka, Bangladesh, 22–24 September 2016; pp. 1–6.
33. Lai, S.; Xu, L.; Liu, K.; Zhao, J. Recurrent Convolutional Neural Networks for Text Classification. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Volume 333, pp. 2267–2273.
34. Shen, Q.; Wang, Z.; Sun, Y. Sentiment Analysis of Movie Reviews Based on CNN-BLSTM. In Proceedings of the International Conference on Intelligence Science, Dalian, China, 23–24 September 2017; pp. 164–171.
35. Yenter, A.; Verma, A. Deep CNN-LSTM with combined kernels from multiple branches for IMDB Review Sentiment Analysis. In Proceedings of the 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), New York, NY, USA, 8–10 November 2017; pp. 540–546.
36. Liu, P.; Qiu, X.; Huang, X. Recurrent neural network for text classification with multi-task learning. *Comput. Lang.* **2016**, arXiv:1605.05101.
37. Chen, T.; Xu, R.; He, Y.; Wang, X. Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Syst. Appl.* **2017**, *72*, 221–230.
38. Wang, X.; Jiang, W.; Luo, Z. Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In Proceedings of the 26th International Conference on Computational Linguistics (COLING): Technical Papers, Osaka, Japan, 11–16 December 2016; pp. 2428–2437.
39. Conneau, A.; Schwenk, H.; Barrault, L.; Lecun, Y. Very deep convolutional networks for text classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Long Papers, Valencia, Spain, 3–7 April 2017; Volume 1, pp. 1107–1116.
40. Wang, J.; Yu, L.C.; Lai, K.R.; Zhang, X. Dimensional sentiment analysis using a regional CNN-LSTM model. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Short Papers), Berlin, Germany, 7–12 August 2016; Volume 2, pp. 225–230.
41. Du, H.; Xu, X.; Cheng, X.; Wu, D.; Liu, Y.; Yu, Z. Aspect-specific sentimental word embedding for sentiment analysis of online reviews. In Proceedings of the 25th International Conference Companion on World Wide Web Conferences Steering Committee, Montreal, QC, Canada, 11–15 April 2016; pp. 29–30.

42. Wang, Y.; Huang, M.; Zhao, L.; Zhao, L. Attention-based lstm for aspect-level sentiment classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 606–615.
43. Poria, S.; Cambria, E.; Gelbukh, A. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowl.-Based Syst.* **2016**, *108*, 42–49.
44. Tang, D.; Qin, B.; Liu, T. Aspect level sentiment classification with deep memory network. *Comput. Lang.* **2016**, arXiv:1605.08900.
45. Zhang, Y.; Gan, Z.; Carin, L. Generating text via adversarial training. *Comput. Lang.* **2016**, *21*, arXiv:1808.08703.
46. Bengio, S.; Vinyals, O.; Jaitly, N.; Shazeer, N. Scheduled sampling for sequence prediction with recurrent neural networks. *Mach. Learn.* **2015**, 1171–1179, arXiv:1506.03099.
47. Yu, L.; Zhang, W.; Wang, J.; Yu, Y. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. *Mach. Learn.* **2017**, 2852–2858, arXiv:1609.05473.
48. Subramanian, S.; Rajeswar, S.; Dutil, F.; Pal, C.; Courville, A. Adversarial generation of natural language. In Proceedings of the 2nd Workshop on Representation Learning for NLP, Vancouver, BC, Canada, 17 January 2017; pp. 241–251.
49. Mogren, O. C-RNN-GAN: Continuous recurrent neural networks with adversarial training. *Artif. Intell.* **2016**, arXiv:1611.09904.
50. Hu, Z.; Yang, Z.; Liang, X.; Salakhutdinov, R.; Xing, E.P. Controllable text generation. *Mach. Learn.* **2017**, arXiv:1703.00955.
51. Semeniuta, S.; Severyn, A.; Barth, E. A hybrid convolutional variational autoencoder for text generation. *Comput. Lang.* **2017**, arXiv:1702.02390.
52. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *Mach. Learn.* **2014**, arXiv:1411.1784.
53. Antipov, G.; Baccouche, M.; Dugelay, J.L. Face aging with conditional generative adversarial networks. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 2089–2093.
54. Lin, K.; Li, D.; He, X.; Zhang, Z.; Sun, M.T. Adversarial Ranking for Language Generation. Advances in Neural Information Processing Systems. 2017. Available online: http://students.washington.edu/kvlin/RankGAN_poster.pdf (accessed on 19 October 2018).
55. Donahue, J.; Krähenbühl, P.; Darrell, T. Adversarial feature learning. *Mach. Learn.* **2016**, arXiv:1605.09782.
56. Makhzani, A.; Shlens, J.; Jaitly, N.; Goodfellow, I.; Frey, B. Adversarial autoencoders. *Mach. Learn.* **2015**, arXiv:1511.05644.
57. Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; Manzagol, P.A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* **2010**, *11*, 3371–3408.
58. Pang, B.; Lee, L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, Barcelona, Spain, 21–26 July 2004; p. 271.
59. Täckström, O.; McDonald, R. Discovering Finegrained sentiment with latent variable structured prediction models. In Proceedings of the European Conference on Information Retrieval, Dublin, Ireland, 18–21 April 2011; pp. 368–374.
60. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. *Comput. Lang.* **2013**, 3111–3119, arXiv:1310.4546.
61. Socher, R.; Lin, C.C.Y.; Ng, A.Y.; Manning, C.D. Parsing Natural Scenes and Natural Language with Recursive Neural Networks. 2011. Available online: https://nlp.stanford.edu/pubs/SocherLinNgManning_ICML2011.pdf (accessed on 19 October 2018).
62. Wang, P.; Xu, J.; Xu, B.; Liu, C.; Zhang, H.; Wang, F.; Hao, H. Semantic clustering and convolutional neural network for short text categorization. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers), Beijing, China, 26–31 July 2015; Volume 2, pp. 352–357.
63. Labutov, I.; Lipson, H. Re-embedding words. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Short Papers), Sofia, Bulgaria, 4–9 August 2013; Volume 2, pp. 489–493.
64. Chollet, F. Keras: Deep Learning Library for Theano and Tensorflow. *Date Sci.* **2015**, *7*, 8.

65. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A System for Large-Scale Machine Learning. *Artif. Intell.* **2016**, *16*, 265–283.
66. Hinton, G.E.; Salakhutdinov, R.R. Replicated softmax: An undirected topic model. In Proceedings of the 23rd Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 7–10 December 2009; pp. 1607–1614.
67. Le, Q.V.; Ngiam, J.; Coates, A.; Lahiri, A.; Prochnow, B.; Ng, A.Y. On optimization methods for deep learning. In Proceedings of the 28th International Conference on Machine Learning, Omnipress, Washington, DC, USA, 28 June–2 July 2011; pp. 265–272.
68. Bernardi, R.; Cakici, R.; Elliott, D.; Erdem, A.; Erdem, E.; Ikizler-Cinbis, N.; Keller, F.; Muscat, A.; Plank, B. Automatic description generation from images: A survey of models, datasets, and evaluation measures. *J. Artif. Intell. Res.* **2016**, *55*, 409–442.
69. Davis, J.; Goadrich, M. The relationship between Precision-Recall and ROC curves. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; pp. 233–240.
70. Sokolova, M.; Japkowicz, N.; Szpakowicz, S. Beyond accuracy, F-score and ROC: A family of discriminant measures for performance evaluation. In Proceedings of the Australasian Joint Conference on Artificial Intelligence, Hobart, Australia, 4–8 December 2006; pp. 1015–1021.
71. Van Hulse, J.; Khoshgoftaar, T.M.; Napolitano, A. Experimental perspectives on learning from imbalanced data. In Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, USA, 20–24 June 2007; pp. 935–942.
72. Saito, T.; Rehmsmeier, M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE* **2015**, *10*, e0118432.
73. Chawla, N.V. Data Mining for Imbalanced Datasets: An Overview. In *Data Mining and Knowledge Discovery Handbook*; Springer: Berlin, Germany, 2009; pp. 875–886.
74. Van Der Maaten, L. Accelerating t-SNE using tree-based algorithms. *J. Mach. Learn. Res.* **2014**, *15*, 3221–3245.
75. Arjovsky, M.; Bottou, L. Towards principled methods for training generative adversarial networks. *Mach. Learn.* **2017**, arXiv:1701.04862.
76. Theis, L.; Oord, A.V.D.; Bethge, M. A note on the evaluation of generative models. *Mach. Learn.* **2015**, arXiv:1511.01844.
77. Fernando, B.; Habrard, A.; Sebban, M.; Tuytelaars, T. Unsupervised visual domain adaptation using subspace alignment. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 2960–2967.
78. LeCun, Y.; Chopra, S.; Hadsell, R.; Ranzato, M.; Huang, F. A tutorial on energy-based learning. In *Predicting Structured Data*; MIT Press: Cambridge, MA, USA, 2006; Volume 1.
79. Denton, E.L.; Chintala, S.; Fergus, R.; others. Deep generative image models using a laplacian pyramid of adversarial networks. *Comput. Vis. Pattern Recogn.* **2015**, 1486–1494, arXiv:1506.05751.