Evaluating Textual Representations through Image Generation

Graham Spinks
Department of Computer Science
KU Leuven, Belgium
graham.spinks@cs.kuleuven.be

Marie-Francine Moens
Department of Computer Science
KU Leuven, Belgium
sien.moens@cs.kuleuven.be

Abstract

We present a methodology for determining the quality of textual representations through the ability to generate images from them. Continuous representations of textual input are ubiquitous in modern Natural Language Processing techniques either at the core of machine learning algorithms or as the by-product at any given layer of a neural network. While current techniques to evaluate such representations focus on their performance on particular tasks, they don't provide a clear understanding of the level of informational detail that is stored within them, especially their ability to represent spatial information. The central premise of this paper is that visual inspection or analysis is the most convenient method to quickly and accurately determine information content. Through the use of text-to-image neural networks, we propose a new technique to compare the quality of textual representations by visualizing their information content. The method is illustrated on a medical dataset where the correct representation of spatial information and shorthands are of particular importance. For four different well-known textual representations, we show with a quantitative analysis that some representations are consistently able to deliver higher quality visualizations of the information content. Additionally, we show that the quantitative analysis technique correlates with the judgment of a human expert evaluator in terms of alignment.

1 Introduction

In this paper, a method is proposed to evaluate the quality of a textual representation by conditioning an image generation network on it.

Neural networks implicitly construct representations of a textual input by learning which features are important for the task at hand. It is not immediately possible however to assess the level of detail and structure that is retained in such a representation. Many systems often complement or replace the input with pre-trained representations that have the advantage of being constructed with a larger unlabeled corpus. Depending on the task, this practice sometimes significantly improves the performance of the network (Turian et al., 2010). On the one hand, this is due to the use of a larger unlabeled corpus which reduces data sparsity and thus improves generalization accuracy. On the other hand, representations often contain higher-level features that are fundamental for the task they are trained for. A neural network in a separate task can thus rely on those features without having to discover them all over again.

As the field of Natural Language Processing advances and machine learning models expand to include multimodal information, the importance of understanding the level of detail and information that is retained in a textual representation only grows. Obtained representations can be employed in additional tasks (for example generation, translation, summarization, etc.) depending on their ability to capture certain types of information. The medical domain in particular might benefit from a better understanding of representations as the industry moves to adopt deep learning methods in increasingly intricate applications and researchers attempt to extract and utilize more complex information structures. An example is spatial information which is an important quantity in many natural language applications, yet no explicit methodology exists that indicates to what extent that information is present in textual representations. In many medical settings, a correct understanding and representation of such information is crucial. In thorax radiography, which is the focus of this paper, textual captions often include detailed findings which relate to specific areas in an X-Ray. Clinical texts in general, add an extra level of com-

Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 30–39
Brussels, Belgium, November 1, 2018.

plexity as they often lack syntactic structure and employ many shorthands.

Images differ from texts in the sense that the retained information and generalization of a representation are immediately apparent for a human observer. It is not surprising that the 'human perceptual score' is a frequently used metric to evaluate image generation systems (Borji, 2018). In this paper we propose a novel method to assess the quality of textual representations. By creating images from different textual representations we show that some representations lack the necessary information to lead to detailed high-quality images. The textual representations are evaluated both by comparing the quality of the produced images compared to the images in the test data, as well as the alignment between images and captions. The outcome is determined both by a qualitative (human perceptual scores) as well as a quantitative (divergence scores) measure. To calculate the divergence scores, we rely on the methodology that estimates distance between two distributions as introduced by (Danihelka et al., 2017) and extend it to estimate how well image and text are aligned in the generated content.

As we show in the results, text-to-image architectures are indeed suitable to get an immediate visual estimate of the quality of the representation and the information contained within. We will evaluate several common textual representations that were constructed with unsupervised learning techniques on both a relatively straightforward conditional GAN as well as on a more advanced StackGan (Zhang et al., 2017) which uses several stages and a conditioning mechanism that augments the textual representation.

The contributions of this paper are:

- The formulation of a methodology to visualize and evaluate the information and quality of different textual representations.
- The extension of a GAN evaluation measure to evaluate alignment of output with conditional information.

2 Motivation and background

To understand the motivation of this paper, it is necessary to understand some background on the different types of textual representations and why better evaluation methods are necessary. As we use text-to-image models for evaluation purposes, we also discuss related research in that area.

2.1 Textual Representations

A textual representation is usually a vector associated with a piece of text, which may be a character, word, sentence, paragraph or document. In its simplest form, a representation can be a symbolic ID, such as in a one-hot vector where each dimension represents an ID. This is essentially a discrete, symbolic representation that is very sparse in information as by definition only one dimension is non-zero. They are also somewhat arbitrary in the sense that two texts that are near each other in the code space don't necessarily share a similar meaning or syntax.

More efficient methods assign particular hand-engineered or automatically extracted features to a lower-dimensional vector. One feature can be stored in exactly one dimension or it could be shared over many. In this paper we will focus on the latter, also referred to as distributed representations or word embeddings, which is the traditional method to represent sentences in recent neural network related research. They are dense, low-dimensional and real-valued (Turian et al., 2010). Texts that contain similar concepts or meaning for a typical task end up near each other in such a distributed representation space which serves as a proxy for generalized, semantic information storage. Word embeddings can be built with unsupervised training, for example by leveraging positional information of texts in a corpus; with weakly supervised training, for example in an adversarial setting; or with supervision of output labels. While this paper focuses on unsupervised and weakly supervised methods only, the methods that are described here are applicable to supervised representations as well.

Well-known methods of creating word embeddings are the word2vec algorithms, introduced by Mikolov et al. (2013a). Word embeddings are usually constructed with neural networks that predict the context of a word in a text document. They are able to scale to large training corpora, thus representing large amounts of information and features in a relatively small amount of dimensions. While word2vec word embeddings solely operate on the word level, extensions have been made that include information at the level of characters (e.g. char-CNN-RNN (Kim et al., 2016)), or at higher

levels such as sentences, paragraphs or documents. (e.g. skipthought vectors (Kiros et al., 2015) or doc2vec (Le and Mikolov, 2014)).

While these methods usually are trained on tasks that reproduce the context of a textual component, autoencoders (AE) are trained to recreate the original text in its entirety while implicitly learning a compact, distributed representation as well of the input text along the way. A recent method that builds on the autoencoder approach is an Adversarially Regularized Autoencoder (ARAE) (Kim et al., 2017). Here, the representation is built explicitly from an encoder that is trained as part of an autoencoder as well as a conventional Generative Adversarial Network (GAN). Such representations contain semantic information about the sentence but also discriminative information that allows the adversarial network to distinguish real samples from fake ones. As a result, a smoother semantic transition is apparent while traversing the representation space when compared to an autoencoder. Spinks and Moens (2018) have applied this technique to create textual representations of X-Ray captions and generate textual output with low perplexity.

The quality of distributed vectors can be assessed with similarity tasks that give a rough measure of semantic and syntactic information (Mikolov et al., 2013a,c) but studies by Faruqui et al. (2016) and Linzen (2016) indeed suggest that the use of word similarity tasks for the evaluation of word vectors is problematic and may lead to incorrect inferences. Schnabel et al. (2015) have evaluated embeddings with a range of methods, both intrinsic, such as semantic and syntactic similarity, and extrinsic, such as noun phrase chunking and sentiment classification. For the extrinsic tasks, they found that different representations performed best for different tasks, suggesting that perhaps there isn't one optimal representation for all tasks. Such studies suggest that better methodologies and more research is needed into methods that accurately assess the value of different continuous representations. This paper addresses this by focusing on the evaluation of the information content of the representation rather than any task-oriented metric. Lazaridou et al. (2015) also worked towards a visualization method for text representations by averaging images of the nearest neighbors vectors after a cross-modal mapping. Contrary to this work, their approach did not include any evaluation mechanism of the outcome and only focused on individual words.

In this paper, we construct distributed representations of sentences with several *unsupervised methods* mentioned above. Subsequently, we propose a new methodology to evaluate the quality of the learned word embeddings by generating images from them, thus visualizing the level of detail and information retained in the different embeddings. To understand our methodology, it is useful to discuss some background on text-to-image models and, more in general, generative models.

2.2 Generative models

Recent text-to-image models rely on advances in generative models, which are probabilistic models that estimate a distribution given a certain input. Such generative systems have shown impressive progress in the creation of realistic data, most notably with Generative Adversarial Networks (GANs) (Goodfellow et al., 2014). In the original formulation, GANs are trained by alternately improving a generator network, G, which aims to create realistic samples and a discriminator network, D, which tries to distinguish real samples from generated ones. As training such an architecture tends to be unstable, several improvements have been proposed, for example the Wasserstein GAN (WGAN) (Arjovsky et al., 2017). In this formulation the discriminator is replaced by a critic, f, that is trained to approximate the Earth-Mover distance (EM). The EM is an estimate of the minimum amount of effort that is necessary to displace one distribution to another (Arjovsky et al., 2017). The loss function to train a GAN with the Wasserstein Distance is presented in Equation 1.

$$\min_G W(G) = \min_G \max_f \mathbb{E}_{x\sim P_r}[f(x)] - \mathbb{E}_{\bar{x}\sim P_g}[f(\bar{x})] \quad (1)$$

where G is the generator, f is the critic, W is the Wasserstein distance, and P_r and P_g are the real and generated data distributions respectively. To ensure that the approximation to the earth mover distance is valid, the critic f should be enforced to be 1-Lipschitz continuous. (Arjovsky et al., 2017) achieve this by clipping the critic weights between $[-c, c]$, where c is typically smaller than 1.

Extensions to the GAN setup have been proposed, such as conditional adversarial networks (Odena et al., 2016), and progressively

grown GANs (Zhang et al., 2017; Karras et al., 2017) which have made detailed high resolution category-dependent image generation possible. During the training of conditional GANs, the class or label is passed along to both generator and discriminator so that the networks implicitly learn relevant auxiliary information which leads to more detailed outputs. Progressively grown GANs rely on low-resolution outputs to learn outlines and structures of images that are refined into smooth visual output at higher resolutions. This approach is also the essence of cross-modal text-to-image architectures. Zhang et al. (2017), for example, have demonstrated how to produce realistic images conditioned on textual captions with a progressive GAN network called StackGAN.

In this paper, we use the StackGAN to visualize textual representations, as well as a simplified text-to-image architecture based on a GAN. The information and quality of the produced images allow us to evaluate the quality of the different textual representations. With that goal we will discuss some methods to evaluate the visual output of such text-to-image GANs.

2.3 Evaluation measures

As we produce images from text to determine the quality of the textual representations, accurate evaluation measures are needed to assess the generated images. We focus on evaluation measures for GANs as it is the only type of architecture that is used to create images in this paper.

Besides human perceptual scores, some recent advances have been made to assess the quality of the distribution of the generated output of GANs. Some of the most widely adopted measures are the Inception Score (IS) (Salimans et al., 2016) and the Fréchet Inception Distance (FID) (Heusel et al., 2017). Both measures have a reasonable correlation with image quality but also contain undesirable properties as explained by Borji (2018). One large problem is that both use a third-party network that was trained on a different dataset to measure the quality of the generated data. It therefore assumes that the distribution of the dataset used in the generation task is similar to the dataset that the third-party network was trained on. This assumption is often not fulfilled, particularly if specialized medical datasets are used.

To solve these issues, Danihelka et al. (2017) propose using divergence and distance functions that are normally used for training a GAN. Im et al. (2018) show that these metrics exhibit consistency across various models and find that they better reflect human perceptual scores than the IS and FID. To calculate how well the generated distribution has approached the data distribution, an independent critic is trained until convergence to distinguish between generated samples and samples from the validation set. The WGAN loss is used and the weights of the original generator are no longer updated. When applied to output images, the Wasserstein distance thus can give an estimate of the divergence between the generated and real images. This quantity is expressed as W_{qual_image} in Equation 2.

$$W_{qual\_image}(G, P_{r,v}) = \max_{f_1}(\mathbb{E}_{x\sim P_{r,v}}[f_1(x)] - \mathbb{E}_{\bar{x}\sim P_g}[f_1(\bar{x})]) \quad (2)$$

where $P_{r,v}$ refers to the real distribution of the validation data.

Additionally, by evaluating the model that is trained in Equation 2 on the training and test set, Danihelka et al. (2017) suggest a method to estimate whether overfitting has occurred. Indeed, if the model generalizes well to the unseen examples in the testset, the expected values in Equation 3 should be roughly the same. In this equation $P_{r,te}$ and $P_{r,tr}$ refer to the real distributions of the test and training set respectively.

$$E[W_{qual\_image}(G, P_{r,te})] = E[W_{qual\_image}(G, P_{r,tr})] \quad (3)$$

While this method allows us to judge the output quality of the images, and by extension the textual representations, in the following section we will explain how our methodology extends this approach in order to evaluate the alignment between image and text.

3 Method

This paper proposes a methodology that evaluates the quality of textual representations by visualizing them with text-to-image models. This is achieved in three separate stages as described in the following subsections.

3.1 Train and create a textual representation

In this paper 4 different textual representations are created by training on the captions of the

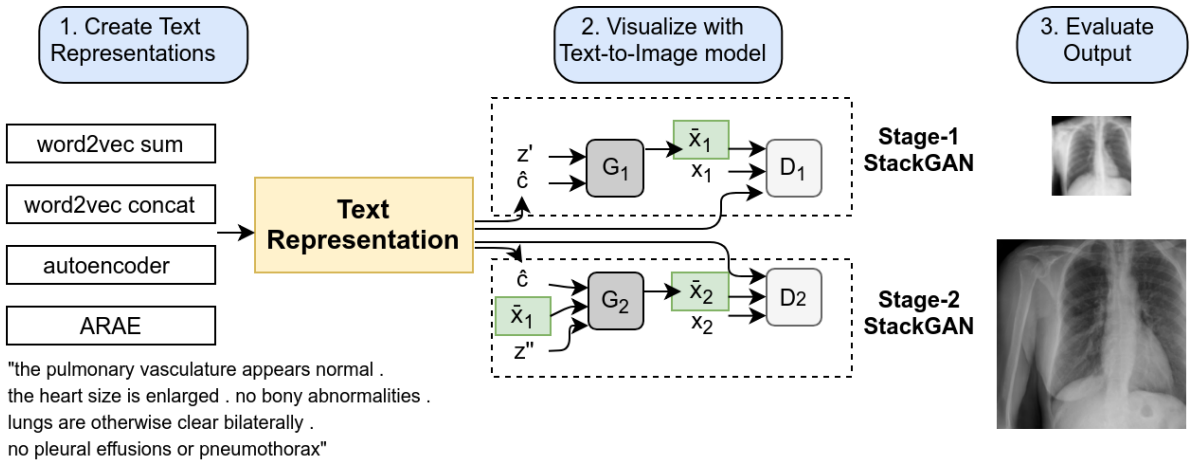

Figure 1. Overview of the methodology. A textual representation is first trained and then fed as a conditional input to a text-to-image model, in this figure a StackGAN. The textual representation is fed to both the first and second stage of an image StackGAN with the goal of creating low- and high-resolution images $\bar{x_1}$ and $\bar{x_2}$ respectively. From the representation, the augmented conditioning embedding $\hat{c}$ is formed. In a final step, the visual output is evaluated.

training set using unsupervised training methods. As these representations are compared afterwards, they each need to have the same, fixed dimension.

For the first 2 representations, the typical word2vec skip-gram word embeddings are used to build the vectors. A representation for a sentence is built by respectively summing and concatenating the individual word embeddings for the entire sequence. Such a comparison is interesting as summing (or averaging) word vectors allows to use high-dimensional word representations, yet sacrifices word order. Concatenating on the other hand, requires the use of low-dimensional word embeddings as the sentence dimension is fixed, but maintains word order and has been shown to work well at the input of convolutional networks (Kim, 2014), such as the text-to-image models used in this paper.

Additionally, the hidden state representation of an autoencoder is built. The autoencoder, that consists of a 1-layer LSTM encoder and a 1-layer LSTM decoder, is trained to recreate the input text with a cross-entropy loss at the word-level.

Finally, we also use the representation produced by an ARAE, as in section 2.1. The ARAE contains a 1-layer LSTM encoder and 1-layer LSTM decoder. The generator and discriminator consist of 3-layer feedforward networks.

3.2 Create images from text

From these representations, images are created with a text-to-image model, which can be a simple conditional GAN or a more complex StackGAN. In the latter, a textual representation t is fed into a fully-connected net that creates a mean μ and a variance σ^2 from which augmented conditional representations $\hat{c}$ are generated. The Kullback-Leibler divergence (KL-loss) is used to coerce $\hat{c}$ to approach a normal distribution $\mathcal{N}(0, I)$. This ensures smoothness between different input texts and avoids overfitting when generating images from captions (Doersch, 2016; Larsen et al., 2015). The conditional vector $\hat{c}$ is then concatenated to a noise vector z', sampled from a normal distribution, and fed to the generator.

Such a StackGAN model is trained in two stages: at a first stage the features of real and generated images are matched to produce low-resolution images that lack detail. During the second stage, the generator produces larger images, conditioned on both the augmented conditional vector $\hat{c}$ as well as the image output of the first stage. The training is broken up into the maximization of the loss of D and the minimization of the loss of G as shown in Equations 4 and 5 for the first stage. Note that a traditional GAN formulation is used in the StackGAN model.

$$\max_{D_1} \mathcal{L}_{D_1} = \mathbb{E}_{x_1 \sim p_d}[log D_1(x_1, t)] + \mathbb{E}_{z \sim p_z, t \sim p_d}[log(1 - D_1(G_1(z, \hat{c}), t))] \quad (4)$$

$$\min_{G_1} \mathcal{L}_{G_1} = \mathbb{E}_{z\sim p_z, t\sim p_d}[log(1 - D_1(G_1(z, \hat{c}), t))] + \lambda D_{KL}(\mathcal{N}(\mu_1(t), \Sigma_1(t))||\mathcal{N}(0, I)) \quad (5)$$

where p_z and p_d represent the random normal and data distribution respectively. t is the textual representation and λ is a regularization parameter to balance the loss between the two terms. Subfix 1 indicates that these equations relate to stage 1.

Note that the StackGAN model is distinct from more conventional text-to-image architectures not only in the sense that the former progressively constructs higher resolution images but also because of the conditioning augmentation. This mechanism is particularly important for this experiment, as it essentially augments the different textual representations. For the simple text-to-image GAN, which we refer to as TTI-GAN, we use a GAN architecture without separate stages that passes the textual representation to both the generator and discriminator without modifications.

Both generator and discriminator for all text-to-image architectures (i.e. the TTI-GAN and both stage-I and stage-II StackGAN) consist of a series of convolutional up- and down-sampling blocks respectively. As the text embedding t is passed to the discriminator it is compressed with a fully-connected network and replicated to match the dimensions of the image.

3.3 Evaluate the output quality

Evaluating the output quality will let us judge the textual representation quality. In order to do so, we can rely on Equation 2 to calculate W_{qual_image}. However, we would also like to have a rough idea of how well the conditional information is assimilated in the output. We therefore extend the previously mentioned setup to calculate the divergence between an additional pair of distributions. $W_{align\_im\_txt}$ in Equation 6 measures the distance between the aligned image-text distributions by also feeding the conditional information, in this case the textual representations, to the critic.

$$W_{align\_im\_txt}(G, P_{r,v}) = \max_{f_2}(\mathbb{E}_{x\sim P_{r,v}}[f_2(x, c)] - \mathbb{E}_{\bar{x}\sim P_g}[f_2(\bar{x}, c)]) \quad (6)$$

where c is conditional information that corresponds to the current data sample. f_2 is distinct and independent from the critic f_1 in Equation 2 but is also trained until convergence on the validation set. The intuition behind Equation 6 is that $W_{align\_im\_txt}$ is a measure of the distance between the real and generated distributions with their conditional information. Thus, $W_{align\_im\_txt}$ should be smaller for models that take the conditional information into account when creating the output.

Note that the value of $W_{align\_im\_txt}$ also depends on the chosen textual representation and can therefore not be used to evaluate alignment of the TTI-GAN model across different representations. It can be used in the case of the StackGAN however as the representations are coerced to approach a normal distribution with the conditioning augmentation mechanism.

We would also like to get an estimate for the amount of overfitting that occurs for each textual representation. For this we rely on the insights of Equation 3. In Equations 7 and 8 we suggest a simple method to compare how much overfitting occurs on both the quality of the images itself, as well as on the alignment to the captions. By taking the quotient of the expected values of the evaluation of W_{qual_image} and $W_{align\_im\_txt}$, we can compare how much overfitting happened for each entity.

$$O_{qual\_image} = E[W_{qual\_image}(G, P_{r,te})]/E[W_{qual\_image}(G, P_{r,tr})] - 1 \quad (7)$$

$$O_{align\_im\_txt} = E[W_{align\_im\_txt}(G, P_{r,te})]/E[W_{align\_im\_txt}(G, P_{r,tr})] - 1 \quad (8)$$

The entire setup of the methodology is illustrated in Figure 1 where the StackGAN architecture is used as the text-to-image architecture.

4 Experiments

The used dataset is the chest X-Ray dataset of the National Library of Medicine, National Institutes of Health, Bethesda, MD, USA (Demner-Fushman et al., 2015). It contains the findings of the frontal and lateral X-Ray for 3851 patients. For this work only the frontal X-Rays are retained. Random crops are performed during training for data augmentation. As the content in the findings is invariant to the order of the sentences, up to 4 captions are created for each X-Ray by selecting different sentences or a different sentence order. Captions that contain less than 30

Representation	W_{qual_image}	σ
w2v sum	0.598	0.033
w2v concat	0.239	0.049
AE (*)	0.243	0.032
ARAE (*)	**0.219**	0.072

Table 1. Quantitative results of 10 runs for the TTI-GAN visualization method for each of the representations. A lower W_{qual_image} implies a better image quality. (*) For both the autoencoder and ARAE, an outlier was removed.

Representation	W_{qual_image}	$W_{align\_im\_txt}$
w2v sum	2.242	**2.239**
w2v concat	2.343	2.360
AE	2.360	2.344
ARAE	**2.229**	2.279

Table 2. Quantitative results for the trained Stage-2 StackGAN visualization method for each of the representations. A lower W_{qual_image} and $W_{align\_im\_txt}$ imply a better image quality and alignment respectively.

words are padded to equal length, with a maximum of 30 words. All words are lowercase and words with a frequency of less than 5 occurrences are removed and replaced by an out-of-vocabulary marker. While the dataset also contains diagnosis labels for each image, they are not used in this paper. The dataset is divided into training, validation and test set with 80%, 10% and 10% of the data respectively.

For the experiments we first create four different textual representations on the captions of the training set, as detailed in section 3.1. Those representations are referred to as word2vec (sum), word2vec (concat), autoencoder and ARAE. To illustrate the methodology, we set the fixed dimension of each representation to 300, which is a standard dimension for such embeddings, initially used by Mikolov et al. (2013b) in their analysis of distributed vectors. For the autoencoder and the ARAE, training is stopped when the validation error of the reconstruction is minimal.

To generate images from the text, the TTI-GAN and StackGAN models are used as explained in section 3.2. The latter produces images with higher resolution than the former approach. This is important as a higher resolution is required to make an accurate assessment about the alignment of the X-Ray images to the captions. The expected outcome is that a textual representation that maintains sequential information performs better than one that does not. Additionally we expect a code that lies on a regularized smooth space, such as the code produced by the ARAE, to be more useful than a code that does not.

Finally, we perform two types of experiments, for which the concrete setup is as follows.

1. As GAN training can be unstable, the TTI-GAN is trained 10 times for each representation. From the evaluation of each, we obtain measures for W_{qual_image}, O_{qual_image} and $O_{align\_im\_txt}$ which allow us to compare the value of the different representations. The TTI-GAN in our setup produces images with a resolution of 64x64 pixels.

2. For the StackGAN, we train one model for each representation, and train an independent critic 5 times for each model. As GAN training can be quite unstable, this experiment does not allow us to judge the value of the representations from just one run. However, we compare our estimates for W_{qual_image} and $W_{align\_im\_txt}$ to the evaluation of a trained clinician, to confirm that our methodology correlates with human judgment, both in terms of quality and alignment. For the first stage of the StackGAN we produce 64x64 pixel images, while the second stage outputs higher resolution 256x256 pixel images. For this experiment, λ was set to 0.05 and c was set to 0.01.

The text-to-image architectures are each trained during 120 epochs for each of the textual representations of the captions in the training set. The image quality is then assessed on the images that are generated from the captions of the validation and test set. This ensures that we check whether the learned representations can generalize well to captions that were never seen during their construction.

5 Results

In Table 1, the quality of the generated images of the TTI-GAN model are presented for each of the representations. Over the ten performed runs, the TTI-GAN training collapsed once for both the

Textual Representation	Results #C/#N	#U
Ground Truth	20/1=20	4
w2v sum	15/4=**3.75**	6
w2v concat	12/8=1.5	5
AE	8/8=1.0	9
ARAE	11/7=1.57	7

Table 3. Qualitative assessment by clinicians for the produced images of the StackGAN Stage-2 model. Are the caption and the image congruent? (Congruent(C)/Not congruent(N)/Unclear(U). Higher values of the proportion #C/#N indicate better alignment.

ARAE and autoencoder representations. As those runs were clear outliers originating from the collapse of GAN training, they were removed from the results in Table 1. As expected, the ARAE results do appear to lead to the best overall image quality, followed by the word2vec (concat) and autoencoder models. The word2vec (sum) consistently leads to worse solutions. In terms of O_{qual_image}, the word2vec (concat) model experiences less overfitting in terms of image quality than the other representations (11.4% versus $15 - 50\%$), suggesting that such concatenated word2vec representations, that maintain word order, generalize well.

While the Stage-2 StackGAN results in Table 2 show that the ARAE representations achieve the highest image quality again, they don't entirely agree with the TTI-GAN results. This can be attributed to several causes: 1. The results for Stage-2 StackGAN only include results for 1 trained model as we would like to compare the metrics for such a model with the human judgment scores; 2. The Stage-2 StackGAN training produces more detailed images of higher resolution so consistent training is more difficult; 3. The augmented conditioning adds to the original representation, likely making the outcome for each representation more similar. With the exception of the autoencoder representation, the outcome of the Stage-2 model, which relies on the outcome of the first stage, exhibit a lot more overfitting in terms of both O_{qual_image} and $O_{align\_im\_txt}$ with values that range from 126% to 498%.

In order to assess the validity of the quantitative assessment, a trained clinician carries out a visual assessment of the produced image samples. We randomly pick 25 produced images of the StackGAN stage-2 models for each of the textual representations. We also selected 25 true caption-image pairs to compare the models to. The evaluator was asked to determine for each sample:

- Are the caption and the generated image congruent or conflicting? (Congruent/ Conflicting/ Unclear)

The evaluator was also asked for each image if it was clearly not a real but generated X-Ray, but didn't find that to be the case for any of the images. This reflects the fact that all W_{qual_image} appear to be quite similar in Table 2. Note that while our model produces an output of 256 by 256 pixels, a higher resolution is still desirable to make accurate judgments about the content of such X-Rays. In cases where the clinician found that additional information would be necessary to judge whether the alignment is correct, the clinician was able to respond with "unclear". Note that this does not mean that the quality of the image was bad.

The results are shown in Table 3. From the results, we find that indeed the word2vec summation model and the ARAE model, that obtained the best alignment scores $W_{align\_im\_txt}$ according to our quantitative measures, also appear to be the best aligned in the human judgment. While the word2vec concatenation model achieved a slightly worse $W_{align\_im\_txt}$ score, the clinician still judged its alignment to be better than the autoencoder model for the selected samples, perhaps reflecting its slightly improved W_{qual_image} over the autoencoder model.

In Figure 1, a generated image of stage-I and stage-II is presented along the architecture. While the Stage-I images capture the structure and main features of the X-Rays, there is a clear improvement in quality for the stage-II images.

6 Conclusion

In this paper, we have proposed a method to determine the quality of textual representations by visualizing them with text-to-image models. After testing our approach on four different unsupervised text-to-image models, it appears that textual representations that retain word order and lie on a smooth representation space, lead to the best quality of image output. We proposed a method to judge the alignment of the captions with the visual output which correlates with the judgment of

a trained clinician. While only unsupervised representations were used in this paper, the methodology can be applied to other types of textual representations. The results in this paper constitute a new methodology to evaluate textual representations through visualization and offer an interesting path for future work. The application of the method to more complex sentences, different fields or topics as well as the development of alternative alignment measures are interesting possibilities for such research.

Acknowledgments

We thank Dr. Erwin Ströker from the UZ Brussel-VUB for sharing his expertise in the qualitative assessment of generated samples.

References

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875*.

Ali Borji. 2018. Pros and cons of gan evaluation measures. *arXiv preprint arXiv:1802.03446*.

Ivo Danihelka, Balaji Lakshminarayanan, Benigno Uria, Daan Wierstra, and Peter Dayan. 2017. Comparison of maximum likelihood and gan-based training of real nvps. *arXiv preprint arXiv:1705.05263*.

Dina Demner-Fushman, Marc D Kohli, Marc B Rosenman, Sonya E Shooshan, Laritza Rodriguez, Sameer Antani, George R Thoma, and Clement J McDonald. 2015. Preparing a collection of radiology examinations for distribution and retrieval. *Journal of the American Medical Informatics Association*, 23(2):304–310.

Carl Doersch. 2016. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*.

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems with evaluation of word embeddings using word similarity tasks. *arXiv preprint arXiv:1605.02276*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a nash equilibrium. *arXiv preprint arXiv:1706.08500*.

Daniel Jiwoong Im, He Ma, Graham Taylor, and Kristin Branson. 2018. Quantitatively evaluating gans with divergences proposed for training. *arXiv preprint arXiv:1803.01045*.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *AAAI*, pages 2741–2749.

Yoon Kim, Kelly Zhang, Alexander M Rush, Yann LeCun, et al. 2017. Adversarially regularized autoencoders for generating discrete structures. *arXiv preprint arXiv:1706.04223*.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, pages 3294–3302.

Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. 2015. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*.

Angeliki Lazaridou, Dat Tien Nguyen, and Marco Baroni. 2015. Do distributed semantic models dream of electric sheep? visualizing word representations through image synthesis. In *Proceedings of the Fourth Workshop on Vision and Language*, pages 81–86.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

Tal Linzen. 2016. Issues in evaluating semantic spaces using word analogies. *arXiv preprint arXiv:1606.07736*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

Augustus Odena, Christopher Olah, and Jonathon Shlens. 2016. Conditional image synthesis with auxiliary classifier gans. *arXiv preprint arXiv:1610.09585*.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307.

Graham Spinks and Marie-Francine Moens. 2018. Generating continuous representations of medical texts. *NAACL HLT 2018*, page 66.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394. Association for Computational Linguistics.

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris Metaxas. 2017. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *IEEE Int. Conf. Comput. Vision (ICCV)*, pages 5907–5915.