Domain-To-Text: improving Domain Generalization using Natural Language

Machine Learning and Artificial Intelligence

Politecnico di Torino

Gaetano Epiro
s277875

Arianna Gentile
s277939

Alessandro Lepori
s280132

Abstract

The ability of a model to work properly on unseen domains is crucial in many realistic scenarios: it is uncommon to know in advance the visual domain of the data with which the network has to work, and it is even more uncommon that it is provided with annotations. Domain Generalization (DG) is a formalization of this scenario: it is composed of several labeled source domains and a single unlabeled target domain, never seen during training. In particular, the sources are characterized by different domain styles, for example art painting, cartoon and sketch, while the target domain will be characterized by a different visual domain, for example photo, and, since the domain shift will be different between different domains, a possible way to improve the classification performance on the target is to use the source visual domain most similar to the one of the target. Another possibility is to weigh the various contributions of visual domains on the basis of their similarity to the target domain. The main objective of this paper is to evaluate how the use of textual descriptions of visual domains can contribute to an improvement in the evaluation of the source-target distance in order to correctly evaluate the contribution of each source domain in the target prediction.

1. Introduction

Nowadays, Machine Learning is widely used in many contexts and areas: its main goal is to allow a model to learn new information from training data, and then apply this model to new test data. Many models rely on the i.i.d. assumption, for which we are assuming that training and testing data are identically and independently distributed, ignoring out-of-distribution scenarios which are commonly encountered in practice. As a consequence of this over-simplified assumption, when the probability distribution of training and testing data are different, the performances of Machine Learning Models drop due to the domain gap. Since collecting data of all possible domains to train ML models is expensive or even impossible, enhancing the generalization ability of a model is fundamental. What we would like is a system that can work uniformly well across multiple data domains, but, since the domains available at training time are often limited, it is common for a model to work well when evaluated on a dataset which belongs to the same populations as the ones of the training data, but do not perform as well when evaluated on different ones with respect to the training data.

The main goal of Domain Generalization (DG) is to learn a system that can perform well across different data distributions, both at training and test time. The system should be able to assimilate as much useful and transferable knowledge as possible from training samples belonging to a limited number of sources, in order to then use this knowledge to classify test samples, which belong to unseen domains. In particular, for this project we used the PACS dataset which features 4 domains, art painting, cartoon, sketch and photo, 3 of them were used during the training phase, while the fourth during the test phase.

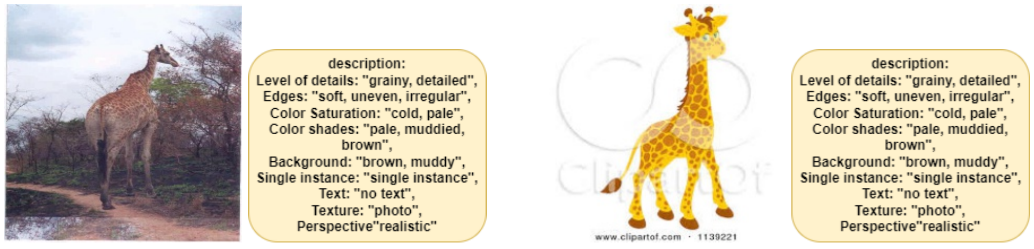
The main goal of our project was to evaluate if the use of textual descriptions in natural language of images belonging to the source domains could be useful in choosing more remarkable weights for the source domains in order to re-evaluate the contributions of them for the target prediction.

While the baselines described in Table 1 were obtained

	ArtPainting	**Cartoon**	**Sketch**	**Photo**	**Avg**
(a)	67.63	57.12	60.40	94.49	69.91
(b)	70.21	57.98	62.03	94.85	71.27

Table 1. Benchmark. (a) were obtained by training 3 models on the sources and evaluating the target. (b) were obtained by using the weights of the DescribingTextures project

Code available at https://github.com/GaetanoEpiro/MLAI_DomainToText



description:
Level of details: "grainy, detailed",
Edges: "soft, uneven, irregular",
Color Saturation: "cold, pale",
Color shades: "pale, muddied, brown",
Background: "brown, muddy",
Single instance: "single instance",
Text: "no text",
Texture: "photo",
Perspective"realistic"

description:
Level of details: "grainy, detailed",
Edges: "soft, uneven, irregular",
Color Saturation: "cold, pale",
Color shades: "pale, muddied, brown",
Background: "brown, muddy",
Single instance: "single instance",
Text: "no text",
Texture: "photo",
Perspective"realistic"

(a) (b) (c) (d)

Figure 1. Example of (image, description) pairs. Figures (a) and (b) define a first (image, description) pair (I,D), figures (c) and (d) define a second (image, description) pair (I',D'), where I' is negative for the description D, while D' is negative for the image I.

by estimating the weights of each domain using the Triplet Match model described in 'Describing Textures using Natural Language' [2] with the dataset called 'DTD2', we randomly chose 400 images from the PACS dataset, and we described each one of them using 9 predefined fields, better described in paragraph 3. All images were used during the training phase, in which we adopted the Metric Learning Approach, described in paragraph 2.

2. Related Work

Self-Supervised Learning.. In recent years, it has been proven that Supervised learning allows to reach extremely good performances if we consider the tasks on which the training was done.

However, it has become known that Supervised learning can be a bottleneck if you want to create generalized models, which work on limited amounts of labeled data. In fact, it is really expensive and time consuming getting a good amount of quality labeled data, especially for more complex tasks, where we need detailed descriptions, but on the other hand for them unlabeled data is already available in great quantities. We can introduce a viable hypothesis that explains how generalized knowledge of the world can be defined as common sense, which is taken for granted as intelligence in human beings. This common sense ability is not innate in Artificial Intelligence, and it is therefore still an open challenge to achieve it. Therefore, the main question is "How can we train a Machine Learning system with a limited amount of training data in order for it to be able to classify correctly the majority of the test data, even if these new data are in an unusual situation, without the help of humans?"

While humans use common sense, which means that they take into consideration previously acquired information as a background knowledge, we need a way to build this background knowledge for ML systems to use it whenever they need it. Self supervised learning enables ML systems to learn useful knowledge of the data from a large quantity of datasets. In our case the labels are available in the data and we use them as an advantage for our system in order to learn more general features.

Today Self supervised learning is used in plenty of tasks, including colorization, 3D rotation, depth completion, or context filling.

Domain Generalization. Domain Generalization (DG) is a Machine Learning problem that consists in considering various scenarios in which the target data is not accessible during the training of a model, unlike for example Domain Adaptation (DA), in which the target data is available during training, even if without labels.

The goal of the DG is to come up with models that are as robust as possible against possible changes to the data distributions in the domains, called domain shifts, starting from the domains of origin available during the training. In particular, a DG model should be able to learn features as general as possible and focus on learning domain-invariant feature representation in order to improve classification on unseen target data as much as possible.

Metric Learning. Many Machine Learning approaches require a measurement of the distance between data points. This distance can be measured with various metrics, such as Euclidean, Cosine, etc. However, these techniques are general and not suitable for particular data or tasks. The metric learning approach allows us to build a distance metric based on the specific task, which can later be used for the main purpose of an activity, such as classification, clustering, etc.

As described in the report 'Describing textures using natural language' [2], the metric learning approach "aims to learn a common embedding over the images and phrases such that nearby image and phrase pairs in the embedding space are related". In particular, metric learning is based on a distance metric which allows us to define how similar or dissimilar the images are to each other, increasing the distance when the images as dissimilar and reducing it if they are similar. In the case of Deep metric learning, Neural networks are used to learn discriminating characteristics be-

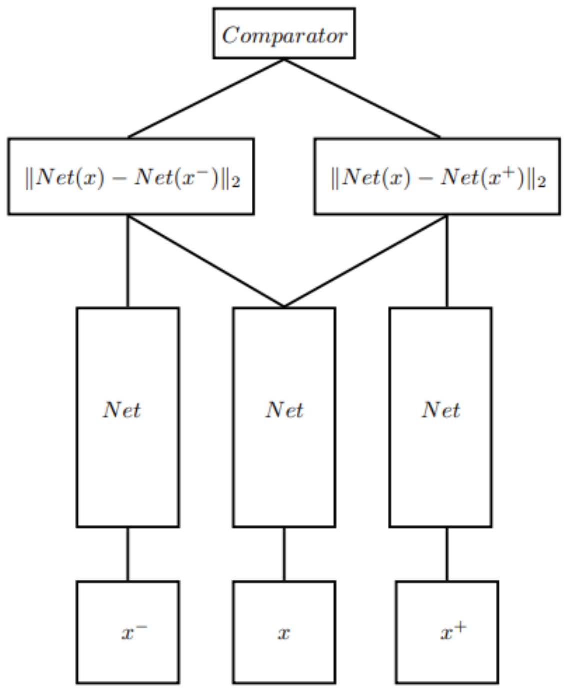


Figure 2. Triplet network structure. [8]

tween the images, and then calculate the metric. The most used loss functions for metric learning are 'contrastive loss' and 'triplet loss', we will focus on the latter.

The triplet network (figure 2) is a neural network where three identical and parallel networks, that share the same architecture and the weights between each other, are trained. Input data is passed through one network in order to define the representation of the distributed embeddings of the data. The main objective is the definition of triplets, formed by an anchor, which is the baseline image, a positive image, which is similar to the anchor image, and a negative image, which is dissimilar to the anchor image. For example, if we take an image from our dataset as the anchor, we can choose as a positive image one that belongs to the same class with respect to the anchor image, and therefore defined as 'positive', while as a negative image we can choose one belonging to a different class, and therefore considered as 'negative'. The cost function for the triplet loss is the following:

$$L(a,p,n) = max(0, D(a,p) - D(a,n) + margin) \quad (1)$$

where D(x,y) is the distance between the learned vector representation of x and y. As distance metric the L2 loss, or the (1-cosine similarity) can be used.

In the 'Describing textures using natural language' project [2], we can consider the embedding of an image ψ(I) and of a phrase φ(P) in $\mathbb{R}^d$. In order to define the triplet we are going to consider for the definition of the loss, we will use as positive a (image, phrase) pair defined as (I, P), and we take as negative a different pair consisting of an image I', negative for the phrase P, and a phrase P' negative for the image I.
We can define two different losses:

$$L_p(I,P,P') = max(0, 1 + ||\psi(I) - \varphi(P)||_2^2 - \\ ||\psi(I) - \varphi(P')||_2^2) \quad (2)$$

L_p from the negative phrase, and

$$L_i(P,I,I') = max(0, 1 + ||\psi(I) - \varphi(P)||_2^2 - \\ ||\psi(I') - \varphi(P)||_2^2) \quad (3)$$

L_i from the negative image.

The main object of this approach is to learn embeddings ψ and of a phrase φ in order to minimize the loss $L = \mathbb{E}_{(I,P),(I',P')}(L_p + L_i)$ over the training set.

For embedding images, the encoder part from ResNet101 is used, in particular activations from layers 2 and 4. An additional linear layer is added with 256 units, resulting in the embedding dimension $\psi(I) \in \mathbb{R}^{256}$.

For embedding phrases (or descriptions) the BERT encoder is used.

BERT. Bidirectional Encoder Representations for Transformers (BERT) was recently introduced in the paper [6] published by researchers at Google AI Language. It is a generic Neural Network model, applied mainly in Natural Language Processing (NLP), able to achieve important results in many tasks, like Question Answering (SQuAD v1.1) or Natural Language Inference (MNLI). The main innovation brought by the BERT encoder is applying the bidirectional training of Transformer, a popular attention model, to language modeling.

Transformers are feed-forward networks (simple layers) and attention blocks (matrices), they include two different mechanism: an encoder that reads the text input, and a decoder which produces predictions for a task, for example, in case we want to generate some text, some probabilities will be generated, in order to indicate the next word to be added to the text. It is important to take into consideration the fact that the Transformer reads the entire sequence of words at once, and therefore it is considered bidirectional. This means that the context of a word is learned based on all of its surroundings (both left and right of the word).

Since BERT is a pre-training model, a multi-level network of encoders is used, in order to learn a representation of the text input (figure 3). As input we can have both a single sentence and a pair of sentences, separated by a separator [SEP] positioned at the beginning and at the end of

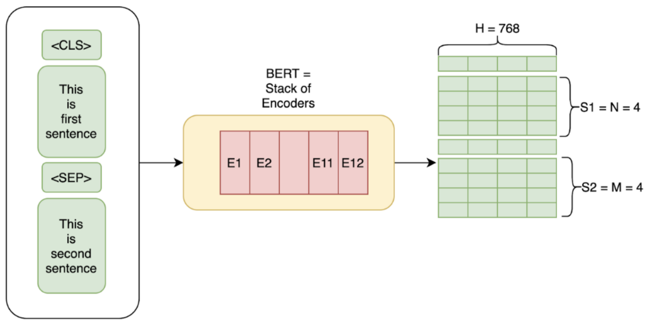


Figure 3. A very High view of BERT — We get a 768 sized vector for all words in our input sentence. [7]

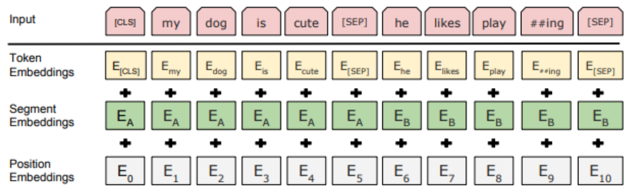


Figure 4. BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings. [6]

the second sentence. Each sentence is converted into sequences of tokens with the wordpiece tokenization, which helps to limit the size of the vocabulary and also helps with out-of-vocab words (for example, the word 'playing' is divided into 'play' and 'ing', and the word 'player' is divided into 'play' and 'er', so that, in case the complete words are not present in the vocabulary, the word 'play' might be). The token embeddings are retrieved by indexing a matrix made of the size of the Vocabulary multiplied by 768(H). We can introduce two different embeddings (figure 4): Segment Embeddings and Position Embeddings. The first kind is used to specify which segment the sentence is from (vectors of H length composed of all 0's if the embedding is from sentence 1, or a vector of 1's if the embedding is from sentence 2), while the latter is used to specify the positions of the words in the sequence.

The main pre-training strategies used by BERT are:

- Masked Language Model: before feeding sequences of words as input to the BERT architecture, some taken randomly words are masked. The model will try to correctly predict the values of these masked words, based on the context of the surrounding words. The main objective of this task is to prevent target words from inadvertently making incorrect connections with other words, as they are all examined at the same time and in their own context.

- Next Sentence Prediction: a pair of sentences are fed to the model, which has to learn to predict if the second sentence is the succeeding of the first one in the original document. During training, 50% of pairs of sentences are made of subsequent sentences, the other 50% are made of a random second sentence.

3. PACS Dataset

For our experiments we randomly extracted a representative subset of 400 images from the PACS dataset, which contains images included in 4 domains (ArtPainting, Photo, Sketch and Cartoon) and 7 common categories ('dog', 'elephant', 'giraffe', 'guitar', 'horse', 'house', 'person').

Annotation. We described the appearance and the style of each image using natural English language. We considered 9 different describable aspects:

- *Level of detail*: If the image is rich in detail or it is a raw representation (e.g. high/mid/low-level)
- *Edges*: Description of the contours of the objects (e.g., definite/precise/neat strokes, definite/precise/neat brush strokes)
- *Color saturation*: The intensity and brilliance of colors in the image (e.g., high/mid/low, vivid colors, light reflections)
- *Color shades*: If there are shades of colors in the image (e.g., colorful/grayscale, etc.)
- *Background*: Description of the background (e.g., monochrome/white/colorful etc.)
- *Single instance*: If the image is composed by a single instance or multiple instances of the same object (e.g., single instance, multiple instances)
- *Text*: If there is text in the picture (e.g., text, no text, dense/sparse text)
- *Texture*: If there is a visual pattern that is repeated over the whole picture (e.g. type of texture)
- *Perspective*: If the three-dimensional proportions of the object parts are realistic (e.g. realistic, unrealistic)

The annotations are expressed as a set of phrases separated by commas (one phrase for each aspect) and each phrase itself is composed of comma-separated sentences.

4. Experiments

The main objective of our project is divided into three sequential steps:

- Reproduction of Simple Ensemble Baseline from the repository [1]

	Art Painting	Cartoon	Sketch	Photo	Avg
(a)	67.63	57.12	60.40	94.49	69.91
(b)	70.21	57.98	62.03	94.85	71.27
(c)	70.51	59.60	63.55	97.31	72.74

Table 2. Final results. (a) simple ensemble baseline. (b) weighted ensemble baseline. (c) final result with weights from finetuned model

- Reproduction of Weighted Ensemble Baseline from the repository [1]
- Finetune the model provided in [2] with the PACS dataset and use the new weights in the starting project [1]

Reproduction of Simple Ensemble Baseline. The first step of our project was to reproduce the baselines described in the first row of the table 1. For the project, 4 different domains were used: ArtPainting, Cartoon, Sketch and Photo, where 3 of them are used as the source, while the fourth is used as the target. In particular, the results for each target domain are obtained by training 3 separate models, each one for each source domain, and taking the mean of the predictions obtained from these models on the target images.

Reproduction of Weighted Ensemble Baseline. In order to reproduce the weighted baselines described in the second row of the table 1, some weights are used to represent the different distances between each source domain and the chosen target domain. The results for each target domain correspond to the weighted mean of the predictions obtained with the models trained on each source domain. For the purpose of obtaining the weights, the model 'Triplet Match' described in [2] is needed (the description of 'Triplet Match' can be found in section 2). The images from target and source domains are encoded in a textual embedding space, where the feature vectors obtained are used in order to compute the source-target distances.

Finetuning the model with the PACS dataset. As already described in section 3, the dataset used in order to finetune the model provided in [2] is made of 400 images randomly chosen from the PACS dataset. With this little dataset and the textual descriptions created by us, we obtained new weights and used them in the starting project [1] in order to obtain new results, described in the table 2. Results in row (c) of table 2 were obtained finetuning our model, using metric learning with Triplet Loss. The weights to train our model were obtained by training the Describing Textures model [4] on the DTD2 dataset and then finetuning it on our dataset, using cosine similarity as distance measure instead of the standard square quadratic distance.

5. Conclusion

As result of using a metric learning approach based on triplet loss, we managed to obtain better accuracy results, showing that adding a textual description using natural human language helps the model to better generalize. Furthermore our results replicated the similarity already present in the ensamble baseline (due to the fact that certain domains are closer with respect to the others, e.g., the domain shift between sketch and photo is larger than the domain shift between cartoon and photo) but our model was able to close even more the domain gap between different domains. Although we used a limited amount of annotated images, which could mean that our model memorized the PACS dataset and consequently overfit on the training samples, we see this work as a possible starting point to a new branch in domain generalization.

References

[1] DomainToText: https://github.com/silvia1993/DomainToText_AMLProject

[2] Chenyun Wu, Mikayla Timm, Subhransu Maji, *Describing Textures using Natural Language*. 2020.

[3] Carlucci, F.M., D'Innocente, A., Bucci, S., Caputo, B., Tommasi, T., *Domain generalization by solving jigsaw puzzles.*. CVPR (2019).

[4] Describing Textures: https://github.com/ChenyunWu/DescribingTextures

[5] Li, Da, et al., *Deeper, broader and artier domain generalization.*. ICCV (2017)

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. Google AI Language, 2019.

[7] James Montantes, *BERT Transformers-How Do They Work?*. Medium, 2021, https://becominghuman.ai/bert-transformers-how-do-they-work-cd44e8e31359

[8] Elad Hoffer, Nir Ailon, *Deep Metric Learning using Triplet Network*. 2018.

[9] DomainToText with DescribingTextures https://github.com/GaetanoEpiro/MLAI_DomainToText