Zero-Shot Learning - An Overview of Classification Problem

Course Project
CS725: Foundations of Machine Learning

TEAM

Mahim Katiha 19i190002
Subhadeep Chaudhuri 19i190010
Saptarshi Majumder 19i190011

Department: Industrial Engineering and Operations Research

Indian Institute of Technology, Bombay
Autumn, 2021-2022

Introduction

Machine learning and deep learning have taken over the world in the last 2 decades, with wide-ranging applications- from object detection to speech recognition to multi-lingual translation and many more. The domain is continuously evolving, with significant progress made over the last 2 decades. Researchers keep breaking the barrier and achieve state-of-the-art performance with new proposed models for various tasks.

In this project, we try to look into the classification task. In particular, we look into this *Zero-Shot Learning* - a problem setup in Machine Learning where a learner is tested over samples from such classes which have not previously been seen by the model (during training), and the learner is posed the challenge to predict the class they belong to.

We set the stage for Zero-Shot learning with a brief overview of its workings in *Sec.1*, the task that we undertake, along with the models and data sets considered in *Sec.2*, the results that we observe in *Sec.3*, and finally we conclude the project in *Sec.4*.

1 Zero-Shot Learning (ZSL)

Let us motivate the significance of Zero-shot learning through an example. Suppose a child knows about horses, but has never seen a zebra. However, the child would have no problem recognizing a zebra if told that a zebra looks very similar to a horse but has black-and-white stripes.

Figure 1: A Horse and a Zebra

(a) Knowing how a horse looks, we might be able to recognize a zebra too if we know that zebra looks like a black-and-white striped horse.

Hence, if we are asked whether we can classify an object without ever seeing it, the answer would be Yes we can, given we have adequate information about its appearance, properties, and functionality.

The foundation of ZSL is combining the observed and unobserved/unseen categories through some types of auxiliary information, which encodes observable distinguishing properties of objects. Thus even without accessing any data of the unseen categories during the training phase, the model is able to transfer intelligence from previously seen categories and auxiliary information. The auxiliary information may include attributes, textual descriptions, or vectors of word category labels. This type of use case is majorly studied in computer vision (CV), natural language processing (NLP), and machine perception.

ZSL is done in two stages:

- **Training**: Where the knowledge about the attributes is captured
- **Inference**: The knowledge is then used to categorize instances among a new set of classes.

How ZSL works?

There are primarily 2 approaches towards the zero-shot recognition problem.

1. **Embedding-based Approach**: The main goal is to map the image features and semantic attributes into a common embedding space using a projection function, which is learned using deep networks.

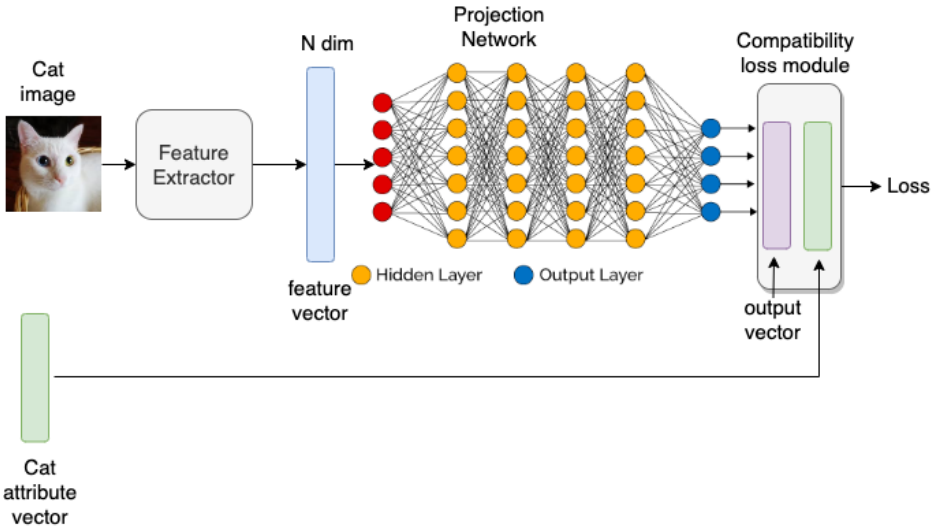

Figure 3: Embedding-based approach to ZSL image classification [1]

In the image provided, the input image is initially passed through a feature extractor network of CNN to get an N-dimensional feature vector for the image. This vector acts as the input to the main network, which returns the result as a D-dimensional output vector. The end goal is to calculate the weights of the projection network so as to map the N-dimensional input to a D-dimensional output. To get this, we put a loss that measures the compatibility between the D-dimensional output and ground truth semantic attribute. The weights of the network are trained such that the D-dimensional output is as close as possible to the ground truth data.

2. **GAN-based Approach**: The main drawback with embedding-based methods is that since the projection function is learned using only seen classes during training, it will be biased towards predicting seen category labels as a result. There is also no surety that the trained projection function will rightly map non-observed category image features to the corresponding semantic space correctly at the testing phase.

   The generative method's goal is to generate image features for non-observed categories using semantic attributes. Generally, this is done using a conditional generative adversarial network (cGAN) that generates image features conditioned on the semantic attribute of a given category.

   Figure 4 depicts the diagram of a general generative model-based zero-shot learning. Identical to the embedding-based method, we use a feature extractor network to get an N-dimensional feature vector. First, the attribute vector is input to the generative model as displayed in the diagram. The generator generates an N-dimensional output vector conditioned on the attribute vector. The generative model is trained such that the synthesized feature vector looks identical to the original N-dimensional feature vector.

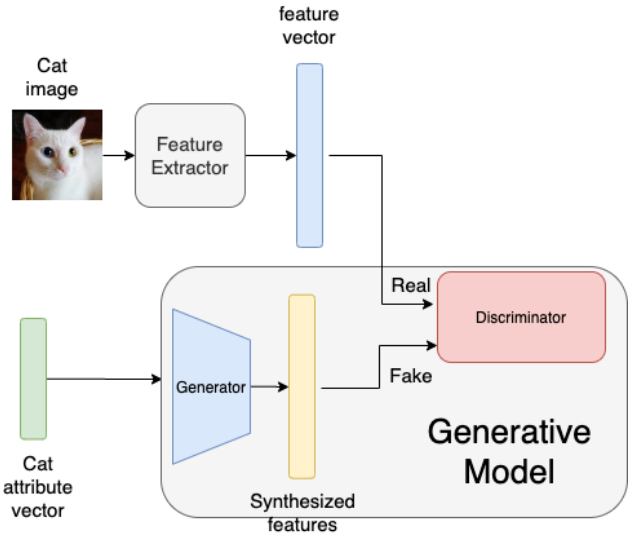

Figure 4: GAN-based approach to ZSL image classification [1]

How ZSL works in text classification?

The zero-shot pipeline in the Transformers library treats text classification as natural language inference (NLI), pioneered by Yin et.al. [2]. In NLI, a model takes two sentences as its input - a premise and a hypothesis - and decides whether the hypothesis follows from the premise (entailment), contradicts it (contradiction), or neither (neutral). let us look at an example for clarity. For the premise "I am deeply in love with you" entails the hypothesis "I like you", is contradicted by "I hate you" and doesn't shed any light or help us draw any conclusions about "She loves me too".

This NLI formulation is used in text classification under the ZSL paradigm by taking the text we'd like to label as the premise, and rephrasing every candidate class as a hypothesis. For a task such as sentiment classification, the premise could be an opinion like "I love this food", with the hypotheses "This sentence is positive", "This sentence is negative" or "This sentence is neutral".

2 Problem Setup - Dataset & Models Considered

Problem Statement

In this part, we demonstrate the Zero-shot learning paradigm in the context of text classification. We namely try to look into datasets of varying class label counts, and multiple models for comparative study of their accuracy. More details about the models and datasets considered are discussed in the following subsections.

Models

We considered 2 models in our study - the Task-Aware Representation of Sentences (TARS) classifier [3] and the Bidirectional and Auto-Regressive Transformer (BART) [4]. We will briefly talk about the 2 classifiers.

1. **TARS classifier**: TARS classifier tries to address a few limitations of the state-of-the-art approaches for text classification. First, the number of classes to predict needs to be pre-defined. In a transfer learning setting, in which new classes are added to an already trained classifier, all information embedded in the linear layer is therefore discarded, and a new layer has to be trained from scratch. Second, this approach only learns the semantics of classes implicitly from training examples, as opposed to leveraging the explicit semantic information provided by the natural language names of the classes.

   Without loss of generality, we can say that the goal of the conventional text classification problem is to find a function:

$$f : text \rightarrow \{0,1\}^M \;\; i.e., \;\; f(t) = P(y_i|t) \;\; \forall i \in \{1, ..., M\}$$

   that maps text (t) to an M-dimensional vector where each dimension (i) corresponds to a particular label (y_i) being either present or not, denoted by probability $P(.)$. For multi-class problems, the labels are mutually exclusive i.e., only one of them can be true. In multi-label settings, multiple labels can be true at the same time for a piece of text. Current state-of-the-art text classification models learn to approximate the function f from task to task, making it infeasible to reuse the existing model for a newer task as outlined earlier.

   To address this challenge, we factorize the text classification problem into a generic binary classification task. Formally, we pose it as a problem of learning a function:

$$f :< task\ label, text > \rightarrow \{0,1\} \;\; i.e., \;\; f(label(y_i), t) = P(\text{True}|y_i, t) \;\; \forall i \in \{1, ..., M\}$$

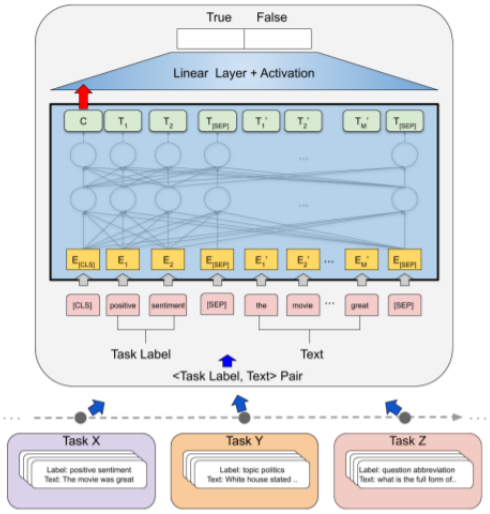

Figure 5: TARS Classifier

   While traditional text classification require one forward pass per task and input to obtain predictions for all M classes, TARS (ref Fig. 6) requires M forward passes, one for each class-input pair.

2. **BART classifier**:

   BART (Bidirectional and Auto-regressive Transformer)is a denoising autoencoder for pretraining seq2seq models. BART is trained by

   (a) Corrupting text tokens with an arbitrarily chosen noise function and

   (b) Learning a model to reconstruct the original text in an auto-regressive manner.

BART uses a standard Transformer architecture (Encoder-Decoder) like the original Transformer model used for neural machine translation but also incorporates some changes from BERT (only uses the encoder) and GPT (only uses the decoder).

BART is pre-trained by minimizing the cross-entropy loss between the decoder output and the original sequence. BART uses the concept of **Masked Language Modeling** (MLM). MLM models such as BERT are pre-trained to predict masked tokens. This process can be broken down as follows:

(a) Replace a random subset of the input with a mask token [MASK]. (Adding noise/corruption)

(b) The model predicts the original tokens for each of the [MASK] tokens

BART has both an encoder (like BERT) and a decoder (like GPT), essentially getting the best of both worlds. The encoder uses a denoising objective similar to BERT while the decoder attempts to reproduce the original sequence (autoencoder), token by token, using the previous (uncorrupted) tokens and the output from the encoder.

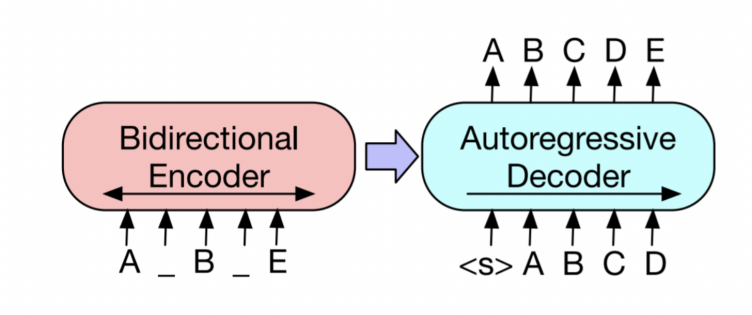

Figure 6: BART Classifier

The corruption schemes used in the paper are summarized below.

(a) **Token Masking** — A random subset of the input is replaced with [MASK] tokens, like in BERT.

(b) **Token Deletion** — Random tokens are deleted from the input. The model must decide which positions are missing (as the tokens are simply deleted and not replaced with anything else).

(c) **Text Infilling** — A number of text spans are each replaced with a single [MASK] token.

(d) **Sentence Permutation** — The input is split based on periods (.), and the sentences are shuffled.

(e) **Document Rotation** — A token is chosen at random, and the sequence is rotated so that it starts with the chosen token.

Data Sets

During our experiments, we used the following 2 datasets:

- ***"Yahoo! Answers topic classification"***: The dataset under consideration is actually constructed from the main *"Yahoo! Answers"* dataset, using 10 largest main categories of the latter. Each class in the dataset contains 140,000 training samples 6,000 testing samples.
- ***"TweetEval"***: TweetEval consists of seven heterogenous tasks in Twitter, all framed as multi-class tweet classification. However, we only consider data for task of the seven tasks - sentiment and emotion.
  - For *"TweetEval-Sentiment"*, we have the following classification labels - negative, neutral and positive which are mapped to integers (0, 1 and 2 respectively) in the dataset.

– For *"TweetEval-Emotion"*, we have the following classification labels - anger, joy, optimism and sadness, which are again mapped to integers (0, 1, 2 and 3 respectively) in the dataset.

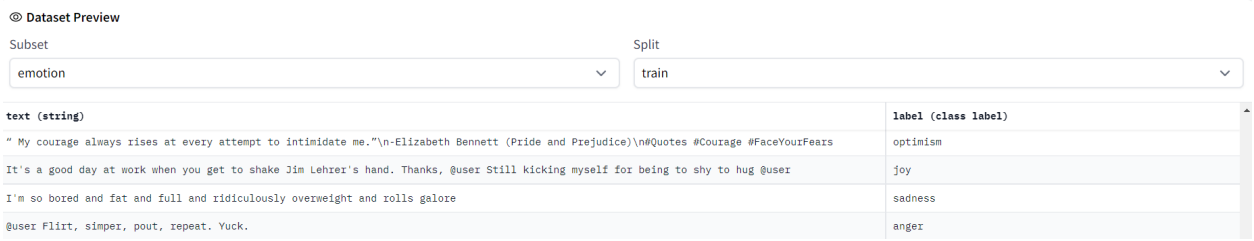

Figure 7: Snippet of "TweetEval-Emotion" dataset [5]

3 Experiments and Results

On analyzing the prospect of Zero-Shot learning over 3 different datasets using our 2 selected models, we came across the following result (shown in Fig. 8) [1].

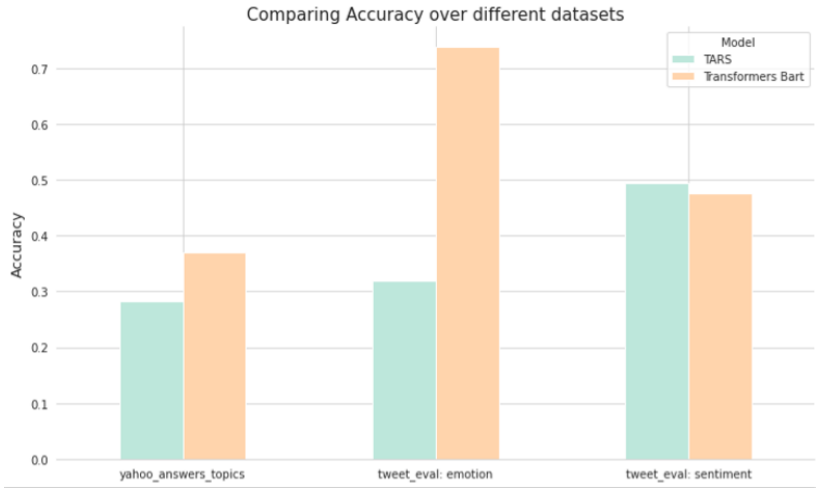

Figure 8: Comparing accuracy of TARS and BART over chosen datasets

The figure shows that BART model performed better than the TARS classifier on 2 out of the 3 datasets. One reason we can attribute this better performance to is the fact that BART uses bidirectional attention mechanism based encoder of BERT and also the autoregressive decoder of GPT simultaneously.

However, one qualm from the observed results lies in the comparative performance of the models over the 2 datasets of very similar domain - *"TweetEval-Emotion"* and *"TweetEval-Sentiment"*. We observe that the BART model performs surprisingly well on the *"TweetEval-Emotion"* dataset, but poorer on the *"TweetEval-Sentiment"* dataset, even though the latter contains lesser number of category labels. A possible explanation in tasks with a default class, such as "neutral", this label can serve as a fallback option, making the model mis-classify many a times. But this seems like a possible area of further study in itself, and would be great to have another set of opinion.

[1] Please find the code and the presentation files in the GitHub repository ***here***

4 Conclusion

In this project, we primarily reviewed the progression and advancements in natural language processing over the last few years, the state-of-the-art models in literature and their novelties (specifically the Transformer architecture with the Attention mechanism, the BERT language model to name a few). We then moved on to investigate an interesting paradigm of transfer learning - the Zero-Shot Leanring, where a learned model predicts classes it has never witnessed previously using certain semantic information about this previously unseen class. We reviewed how the ZSL paradigm works, and tried experimenting ourselves on the task of text classification. We considered multiple datasets and underlying baseline models for our study and reported the accuracy of each of these models.

Acknowledgement

We would like to thank our course instructor Prof. Preethi Jyothi for this course, especially the way she had always been there for the students and creating a safe space for us, inside and outside class hours. A special thanks to the TAs too, who had been amazing in resolving our doubts and issues thrughout.

References

[1] "Zero-shot learning." `https://www.kdnuggets.com/2021/04/zero-shot-learning.html`.

[2] W. Yin, J. Hay, and D. Roth, "Benchmarking Zero-shot Text Classification: Datasets, Evaluation, and Entailment Approach," in *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

[3] K. Halder, A. Akbik, J. Krapac, and R. Vollgraf, "Task-aware representation of sentences for generic text classification," in *Proceedings of the 28th International Conference on Computational Linguistics*, (Barcelona, Spain (Online)), pp. 3202–3213, International Committee on Computational Linguistics, Dec. 2020.

[4] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (Online), pp. 7871–7880, Association for Computational Linguistics, July 2020.

[5] "TweetEval - Emotion dataset." `https://huggingface.co/datasets/tweet_eval/viewer/emotion/train`.

[6] "Zero-Shot Learning - Dr. Timothy Hospedales, Yandex School of Data Analysis Conference." `https://youtu.be/jBnCcr-3bXc`.

[7] "Zero-Shot Classification with HuggingFace Pipeline." `https://youtu.be/J6D-S9gfgwk`.

[8] "Zero-Shot Learning in modern NLP, Joe Davison Blog." `https://joeddav.github.io/blog/2020/05/29/ZSL.html`.