Economics 2355: Zero-Shot and Few-Shot Learning in NLP

Harvard University

April 2021

Outline

What It Means to Learn in Just a Few Shots

Few-Shot Learning and Zero-Shot Learning in Practice

A Motivation for Zero- and Few-Shot Learning

- Up until this point, we've mostly discussed the tremendous success of neural language models (e.g., BERT) as applied to specific tasks (e.g., text classification, question answering) when they are *explicitly fine-tuned* to do very well on these tasks
- In some sense, this comes along with the territory of NLP being a very benchmark-focused field: benchmark datasets often beget benchmark tasks, so, almost by definition, models that hope to do well on benchmark tasks will have a large collection of data from which to train supervised models

A Motivation for Zero- and Few-Shot Learning

- Take SQuAD for example: every SOTA result that has been set on SQuAD has been able to use over 100K question/answer pairs for fine-tuning a model
- However, as we saw last lecture, models trained on one dataset may have trouble generalizing to others
- Or, as the paper that introduced GPT-3 puts it: “to achieve strong performance on a desired task typically requires fine-tuning on a dataset of thousands to hundreds of thousands of examples specific to that task. Removing this limitation would be desirable, for several reasons.”

A Motivation for Zero- and Few-Shot Learning

- Some of these reasons include:
  - For any non-benchmark task of interest, we need to assemble another large, task-specific labeled dataset to get comparable SOTA results, i.e., “the need for a large dataset of labeled examples for every new task limits the applicability of language models”–this is especially salient in low-resource settings
  - Models fine-tuned on a large dataset for one task tend to generalize poorly on other, even similar tasks, i.e., “generalization achieved under this paradigm can be poor because the model is overly specific to the training distribution”
  - The task-specific fine-tuning paradigm inhibits the dream of true AGI, i.e., “humans do not require large supervised datasets to learn most language tasks”

A Motivation for Zero- and Few-Shot Learning

- In other words, even with the architectural advances of the last few years, state of the art performance requires state of the art data
- For any and all of the above reasons, researchers have started to care more and more about what are called “zero-shot,” “one-shot,” and “few-shot” NLP methods and models

An Informal Definition of Zero- and Few-Shot Learning

- The exact definition of few-shot learning (FSL) changes with the context in which it is applied, but, in general, FSL refers to a paradigm in which a ML/DL model only requires a *few* labeled examples to achieve satisfactory performance on some novel task of interest
    - One-shot learning (1SL) is FSL in a context where only one labeled example is needed to achieve sufficient performance on some novel task of interest
    - Zero-shot learning (0SL) is FSL in a context where *no* labeled examples are needed to achieve sufficient performance on some novel task of interest

An Informal Definition of Zero- and Few-Shot Learning

- For example, 0SL in the context of text classification would imply that a model can learn arbitrary task-specific classes for arbitrary texts without any task-specific supervised training/fine-tuning
- More broadly, FSl, 1SL, and 0SL can be thought of as getting a model to do something that it wasn't explicitly trained to do (Davison 2020), or getting a model to perform some previously unencountered downstream task without any parameter or architecture modification (Radford et al. 2019, a.k.a. the GPT-2 paper)
- FSL, 1SL, and 0SL stand are the antithesis of benchmark chasing

An Intuition for Zero- and Few-Shot Learning

- How, though, can a model do something that it wasn't explicitly trained to do?
- Different ML subfields have different approaches to answering this question, but one of the most common in NLP is, simply put, transfer learning!
- Or: models trained to do some general purpose NLP tasks well on lots of diverse data may be good at performing other previously unencountered tasks

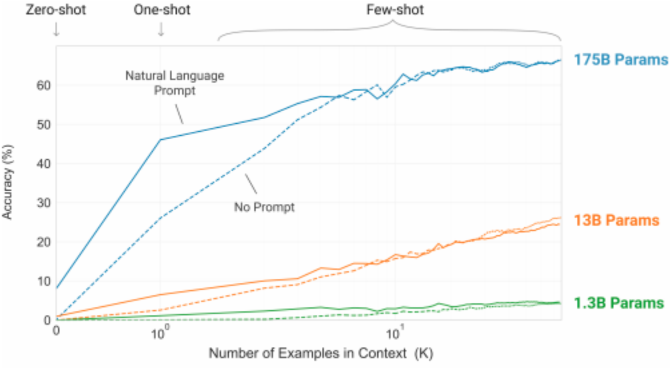
The Poster Child for FSL for NLP: GPT-3

- A case in point is GPT-3
- The paper from OpenAI that introduced GPT-3 was called “Language Models are Few-Shot Learners,” and for good reason
- Essentially, researchers at OpenAI found that by training an enormous, 175B parameter neural language model (a deep transformer-decoder stack) with the right sort of pre-training objective on a massive, 300B token corpus, it was made capable of performing various downstream NLP tasks at a high level without ever having been explicitly trained or fine-tuned to perform them

The Poster Child for FSL for NLP: GPT-3

- The punchline of the paper is that, using just a simple transformer-decoder architecture, knowledge acquired through lots of language modeling can be transferred very effectively to the performance of non-language-modeling tasks
- Moreover, with more and more parameters and pre-training you can push a neural language model so far that it “achieves promising results in the zero-shot and one-shot settings, and in the the few-shot setting is sometimes competitive with or even occasionally surpasses state-of-the-art” (Brown et al. 2020)

The Poster Child for FSL for NLP: GPT-3



The Tradeoff of FSL

- While these results are impressive, it is important to note that even a model as big as GPT-3 can't get SOTA performance on many downstream tasks, for which models fine-tuned on task-specific datasets are still the best performers
- This is the tradeoff inherent in FSL models: although you benefit from not needing to assemble a large task-specific dataset for fine-tuning, you endure the cost of not (usually) being able to get the task-specific performance that is possible through fine-tuning, all else equal
- It's therefore important to keep your use case (and a concept of "sufficient" performance for that use case) in mind when contemplating whether or not to use a FSL model or method

Outline

What It Means to Learn in Just a Few Shots

Few-Shot Learning and Zero-Shot Learning in Practice

FSL and 0SL in Practice

- Despite GPT-3's hype and demonstrated success as a few-shot and zero-shot learner, it is not a model you will use in practice for FSL, e.g., because of restrictions to its availability, the tons of compute needed to even run inference with the model, and the lack of transparency surrounding the model's maintainence and development
- That said, outside of GPT-3, there are still many powerful, open source FSL and 0SL models that you may find useful in practice

Classification as Natural Language Inference

- One of the best resources for doing FSL and 0SL in practice comes from a blog post written by Huggingface researcher Joe Davies: `https://joeddav.github.io/blog/2020/05/29/ZSL.html`
- Of interest to many will be his primer on framing "Classification as Natural Language Inference"

Classification as Natural Language Inference

- The idea behind framing classification as natural language inference is to use “a pre-trained MNLI sequence-pair classifier as an out-of-the-box zero-shot text classifier,” as pioneered in Yin et al. (2019)
- Natural language inference (NLI) is a task that takes as input a “premise” and “hypothesis” sentence and performs three-way classification as to whether the given premise “entails,” “contradicts,” or is “neutral” with respect to the hypothesis
- If one takes an arbitrary piece of text to be a premise, and takes a sentence declaring that text as a certain topic to be a hypothesis, then NLI entailment would correspond to that arbitrary piece of text being declared a certain topic with some softmaxed probability, i.e., topic classification!

Classification as Natural Language Inference

With the Huggingface `transformers` API, this can be accomplished as:

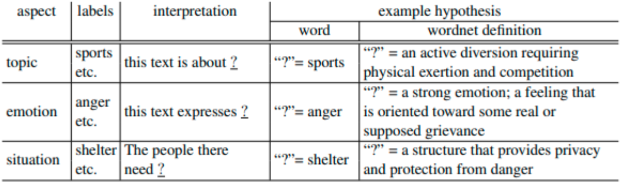
```python
# load model pretrained on MNLI
from transformers import BartForSequenceClassification, BartTokenizer
tokenizer = BartTokenizer.from_pretrained('facebook/bart-large-mnli')
model = BartForSequenceClassification.from_pretrained('facebook/bart-large-mnli')

# pose sequence as a NLI premise and label (politics) as a hypothesis
premise = 'Who are you voting for in 2020?'
hypothesis = 'This text is about politics.'

# run through model pre-trained on MNLI
input_ids = tokenizer.encode(premise, hypothesis, return_tensors='pt')
logits = model(input_ids)[0]

# we throw away "neutral" (dim 1) and take the probability of
# "entailment" (2) as the probability of the label being true
entail_contradiction_logits = logits[:,[0,2]]
probs = entail_contradiction_logits.softmax(dim=1)
true_prob = probs[:,1].item() * 100
print(f'Probability that the label is true: {true_prob:0.2f}%')
```

And now even:

```python
from transformers import pipeline
classifier = pipeline("zero-shot-classification")
sequence = "Who are you voting for in 2020?"
candidate_labels = ["politics", "public health", "economics"]
classifier(sequence, candidate_labels)
```

Classification as Natural Language Inference

- Huggingface also provides a live demo of the classifier here, which could prove valuable for quickly testing out how well this method might work for your use case: `http://35.208.71.201:8000/`
- It’s also worth noting that, for this NLI based zero-shot text classifier, the way the declaration of a topic is made–the “hypothesis template”–could have a strong effect on your results; in Yin et al. (2019), they employ the following templates for various classification tasks:



aspect	labels	interpretation	example hypothesis	
			word	wordnet definition
topic	sports etc.	this text is about ?	“?”= sports	“?” = an active diversion requiring physical exertion and competition
emotion	anger etc.	this text expresses ?	“?”= anger	“?” = a strong emotion; a feeling that is oriented toward some real or supposed grievance
situation	shelter etc.	The people there need ?	“?”= shelter	“?” = a structure that provides privacy and protection from danger

Table 4: Example hypotheses we created for modeling different aspects of 0SHOT-TC.

Classification as Natural Language Inference

- Classification as natural language inference is just an example of transfer learning facilitating 0SL: when text inputs are cleverly structured, natural language understanding jointly learned from MLM pre-training and NLI fine-tuning allows for a neural language model to perform topic classification without any actual topic classification-specific fine-tuning!

Classification as Natural Language Inference

- The default Huggingface zero-shot text classifier uses the FAIR model BART as fine-tuned on the MultiNLI (MNLI) dataset as its neural language model
- A note on BART:
  - BART is a model that closely resembles the original transformer architecture in that it has an encoder and decoder stack
  - BART frames language modeling pre-training as a sequence-to-sequence task, i.e., BART is trained by "(1) corrupting text with an arbitrary noising function [on the encoder side], and (2) learning a model to reconstruct the original text [on the decoder side]"
  - In practice, the authors of BART find that noising sentences by randomly shuffling sentence order and replacing spans of text with [MASK] tokens can lead to performance superior to RoBERTa on some tasks
  - By contrast, a model like BERT only has one noising function, which is replacing some percentage of individual tokens with [MASK] tokens

Classification as Natural Language Inference

- At the time the Huggingface zero-shot classifier was created, the availability of pre-trained BART models and BART’s SOTA or near-SOTA performance on many NLP tasks including NLI made it a convenient choice of underlying neural language model, but because BART is just one example of a model that can be put into “sequence classification” mode, it can be easily swapped out in the zero-shot setting, e.g., with Big Bird for classifying longer sequences of text, with RoBERTa pre-trained on a larger news corpus for tasks in the domain of news

Classification as Natural Language Inference

- To see the Huggingface zero-shot classifier in action for two distinct use cases, visit: `https://tldrstory.com/`
- As a word of caution, the blog author also notes that: “Fine-tuning this model on a small number of annotated data points is not effective, so it is not particularly amenable to the few-shot setting”

Classification as a Cloze Task

- Another promising few- or zero-shot approach to text classification is based on the work of Schick et al. (2020) and their “Pattern Exploitation Training” (PET) method
- The basic idea behind PET is that if you have enough unlabeled training data for your use case, you can create a high performing text classifier by training it on “soft labels” (i.e., high confidence, predicted labels)

Classification as a Cloze Task

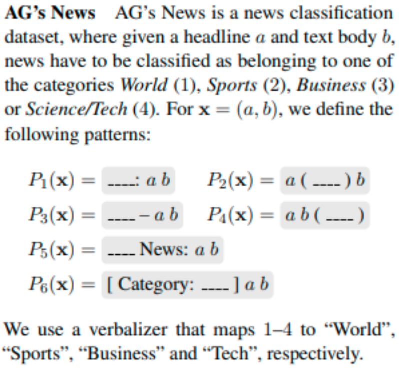
- To get these soft labels, we can dress up texts in a “cloze” pattern, and, similarly to MLM pre-training, have a neural language model guess a completion to the cloze pattern that corresponds to a label of interest
- E.g., “Apples are the color [MASK]”; here “[MASK]” might be filled in with “red” with a high probability, corresponding to a color classification task, and the cloze pattern would simply be “[INPUT] are the color [MASK]”

Classification as a Cloze Task

- We can manually create several of these cloze patterns (e.g., “[INPUT] are the color [MASK]” and “The color of [INPUT] is mostly [MASK]”) and, for each, assuming we have or can make a few ground truth labeled examples, fine-tune a neural language model for filling in “[MASK]”
- We can then have all of these cloze pattern fine-tuned models vote on the best fill-in/label for the full universe of unlabeled training data we have access to, and then consider these to be soft labels for training a text classifier in a proper supervised fashion
- Importantly, this method assumes that you have lots of data for unsupervised training before FSL-based inference can begin

Classification as a Cloze Task

A real world example of framing classification as a cloze task (using multiple cloze patterns as in the previously described algorithm) is the following, taken directly from Schick et al. (2020):



AG's News AG's News is a news classification dataset, where given a headline a and text body b, news have to be classified as belonging to one of the categories *World* (1), *Sports* (2), *Business* (3) or *Science/Tech* (4). For $\mathbf{x} = (a, b)$, we define the following patterns:

$$P_1(\mathbf{x}) = \boxed{\underline{\quad}:\ a\ b} \qquad P_2(\mathbf{x}) = \boxed{a\ (\ \underline{\quad}\ )\ b}$$

$$P_3(\mathbf{x}) = \boxed{\underline{\quad} - a\ b} \qquad P_4(\mathbf{x}) = \boxed{a\ b\ (\ \underline{\quad}\ )}$$

$$P_5(\mathbf{x}) = \boxed{\underline{\quad}\ \text{News:}\ a\ b}$$

$$P_6(\mathbf{x}) = \boxed{[\ \text{Category:}\ \underline{\quad}\ ]\ a\ b}$$

We use a verbalizer that maps 1–4 to "World", "Sports", "Business" and "Tech", respectively.

Classification as a Cloze Task

- Schick et al. (2020) also provide a method called iPET that can be used in a true zero-shot capacity–the authors report performance for iPET on par with some supervised text classification methods
- Just as with the NLI-based zero-shot classifier, PET is finding a way to extract knowledge buried within a pre-trained neural language model to generate text classifications

TANDA as a Cousin of FSL

- We've now learned that FSL and 0SL methods frequently make use of off-the-self models with training that built up general or task-relevant knowledge–be it from pre-training or fine-tuning–and exploit this knowledge in such a way that novel tasks can be performed with little or no additional novel task-specific labeled data
- The paper "TANDA: Transfer and Adapt Pre-Trained Transformer Models for Answer Sentence Selection" (Garg et al. 2019) stretches this insight further, empirically demonstrating that if "we first transfer a pre-trained model into a model for a general task by fine-tuning it with a large and high quality dataset" and then we "perform a second fine-tuning step to adapt the transferred model to the target domain," it is possible to get SOTA performance on many NLP tasks, e.g., question answering

TANDA as a Cousin of FSL

- While TANDA is not described as a FSL method by its authors, it can be thought of as closely related: by exploiting fine-tuning on large labeled datasets close to a task of interest to you, or models already fine-tuned on these datasets, you may only need a few (or far fewer) labeled examples to achieve satisfactory performance on your use case specific task
- Once again, transfer learning can be seen as the bedrock of good FSL

Is FSL or 0SL for You?

- Although FSL and 0SL involve fundamental tradeoffs between data collection costs and model performance, for many use cases FSL and 0SL models and methods may provide an optimal balance between accuracy and effort
- Fortunately, the very nature of 0SL/FSL–coupled with well developed, open source code from authors like Huggingface–makes it very easy to assess whether or not 0SL or FSL works for your use case
- FSL and 0SL is also for more than just pure NLP: recently, OpenAI unveiled a new model called CLIP that is capable of doing high quality zero-shot classification for image and text problems, which may also prove helpful if your use case involves any sort of image labeling