Continual Few-Shot Learning for Text Classification

Ramakanth Pasunuru[1,2] **Veselin Stoyanov**[2] **Mohit Bansal**[1]
[1]UNC Chapel Hill
[2]Facebook AI
{ram,mbansal}@cs.unc.edu, ves@fb.com

Abstract

Natural Language Processing (NLP) is increasingly relying on general end-to-end systems that need to handle many different linguistic phenomena and nuances. For example, a Natural Language Inference (NLI) system has to recognize sentiment, handle numbers, perform coreference, etc. Our solutions to complex problems are still far from perfect, so it is important to create systems that can learn to correct mistakes quickly, incrementally, and with little training data. In this work, we propose a continual few-shot learning (CFL) task, in which a system is challenged with a difficult phenomenon and asked to learn to correct mistakes with only a few (10 to 15) training examples. To this end, we first create benchmarks based on previously annotated data: two NLI (ANLI and SNLI) and one sentiment analysis (IMDB) datasets. Next, we present various baselines from diverse paradigms (e.g., memory-aware synapses and Prototypical networks) and compare them on few-shot learning and continual few-shot learning setups. Our contributions are in creating a benchmark suite[1] and evaluation protocol for continual few-shot learning on the text classification tasks, and making several interesting observations on the behavior of similarity-based methods. We hope that our work serves as a useful starting point for future work on this important topic.

1 Introduction

Large end-to-end neural models are becoming more pervasive in Computer Vision (CV) and Natural Language Processing (NLP). In NLP in particular, large language models such as BERT (Devlin et al., 2019) fine-tuned end-to-end for a task, have advanced the state-of-the-art for many problems such as classification, Natural Language Inference (NLI), and Question Answering (QA) (Devlin et al., 2019; Liu et al., 2019; Wang et al., 2019). End-to-end models are conceptually simpler than the previously-popular pipelined models, making them easier to deploy and maintain. However, because large end-to-end models are black-boxes, it is difficult to correct the mistakes that they make. Practical, real-world applications of NLP require such mistakes to be corrected on the fly as the system operates. For example, when a translation system makes a harmful mistake (e.g., translates "EMNLP" to "ICML"), a phrase-based system can be corrected by finding and modifying the responsible entries in the phrase table (Zens et al., 2002), whereas there is no equivalent way to correct that in an end-to-end neural MT system. Similarly, systems have been shown to exhibit bias (e.g., gender or racial stereotypes) toward certain inputs of text, which we want to correct via few examples on the fly.

Further, the examples that provide supervision to correct mistakes or learn a phenomenon are often hard or impossible to acquire (e.g., due to privacy or ethics issues) (Wang et al., 2020). Hence, it is important to effectively learn to correct mistakes using few extra training examples. Recent work has shown the generalization capability of large pre-trained models to handle multiple tasks with zero to few training examples (Schick and Schütze, 2021; Brown et al., 2020; Yin et al., 2020). For example, Yin et al. (2020) has shown that system trained for NLI can be used to perform new tasks zero-shot, i.e., without any task-specific training data. We believe that similar models can be used to rapidly learn to correct a phenomenon within the same task from a few (e.g., 10 or 15) training examples.

From a practical point of view, we need our trained systems to rapidly adapt to new phenomena (or correct its mistakes) using very few extra training examples, and do it continually as new phenomena (or errors) are discovered over time.

[1]https://github.com/ramakanth-pasunuru/CFL-Benchmark

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 5688–5702
November 7–11, 2021.

Tackling this important setting, we take a fresh look at continual learning in NLP and formulate a new setting that bears similarity to both continual and few-shot learning, but also differs from both in important ways. We dub the new setting "continual few-shot learning" (CFL) and formulate the following two requirements:

1. Models have to learn to correct classes of mistakes (or adapt to new domains) from only a few examples.
2. They have to maintain performance on previous test sets.

To this end, we propose a benchmark suite and evaluation protocol for continual few-shot learning (CFL) on text classification tasks. Our benchmark suite consists of both existing and newly created datasets. More precisely, we use the dataset with several linguistic categories annotated by Williams et al. (2020) from ANLI Round-3 (Nie et al., 2020); and also provide two new datasets with linguistic categories that we annotated using the counterfactual augmented data provided by Kaushik et al. (2020) on SNLI natural language inference dataset (Bowman et al., 2015) and IMDB sentiment analysis dataset (Maas et al., 2011).

We discuss several methods as important promising baselines for CFL, borrowing from the literature of few-shot learning and continual learning. We classify these baselines into parameter correction methods (e.g., MAS (Aljundi et al., 2018)) and non-parametric feature matching methods (e.g., Prototypical networks (PN) (Snell et al., 2017)). We compare these methods on our benchmark suite in a traditional few-shot setup and observe that non-parametric feature matching methods perform surprisingly better than other methods. Next, we test the same methods in a continual few-shot setup and observe that a simple fine-tuning method performs better than other parameter correction methods like MAS. The non-parametric feature matching based PN performs well on the examples that are being corrected (few-shot categories), but at the expense of the original performance. Further, we also observe a large performance improvement on the few-shot categories in this setup. Additionally, we provide interesting ablations to understand the usefulness and generalization capabilities of PN for few-shot linguistic categories. We compare models trained with cross-entropy loss versus Prototypical loss via empirical studies and t-SNE plots, and discuss their major differences in detail. We hope that our CFL benchmark suite and evaluation protocol will serve as a useful starting baseline point and encourage substantial progress and future work by the community on this important practical setting.

2 Related Work

CFL bears similarity to few-shot learning, continuous learning, and online learning. Below, we discuss these three paradigms and highlight the similarity and differences from our approach.

Few-Shot Learning. The goal in few-shot learning is to learn a new task from only a few labeled examples. Few-shot learning problems are studied in the image domain (Koch et al., 2015; Vinyals et al., 2016; Snell et al., 2017; Ren et al., 2018; Sung et al., 2018), focusing mainly on two kinds of approaches: metric-based approaches and optimization-based approaches. Metric-based approaches learn generalizable metrics and corresponding matching functions from multiple training tasks with limited labels (Vinyals et al., 2016). For example, Snell et al. (2017) proposed to build representations for each class using supporting examples and then comparing the test instances by Euclidean distances. Optimization approaches aim to learn to optimize model parameters based on the gradients computed from limited labeled examples (Ravi and Larochelle, 2017; Munkhdalai and Yu, 2017; Finn et al., 2017).

In the language domain, Yu et al. (2018) proposed to use a weighted combination of multiple metrics obtained from meta-training tasks for inferring on a newly-seen few-shot task. On the dataset side, Han et al. (2018) introduce a few-shot relation classification dataset. Recently, large-scale pre-trained language models have been used for few-shot learning of downstream tasks (Brown et al., 2020; Schick and Schütze, 2021). Yin et al. (2020) used pre-trained entailment system for generalizing across more domains or new tasks when there are only a handful of labeled examples.

All of the above-mentioned approaches focus on few-shot learning for new tasks. In contrast, we consider the same original task, but target examples that can be considered new because they require solving a linguistic phenomenon, an error category, or a new domain. Unlike few-shot learning, we also require models that can maintain or improve performance on the existing data.

Continual Learning. Continual learning is a long-standing challenge for machine learn-

Dataset	Categories	Example
ANLI R3	Numerical, Reference	**Context:** Police said that a 21-year-old man was discovered after he had been shot in South Jamaica on Aug. 18 and is in critical condition. Just before 9:30 p.m., police responded to a shooting at 104-46 164th St and discovered the victim, whose name has not been released, at the scene. The victim was shot in the thigh and transported to Jamaica Hospital, where he is currently listed in critical condition. No arrests have been made in the incident. **Hypothesis:** The victim was less than a quarter century old. **Label:** Entailment
IMDB	Negation	**Original Text:** We know from other movies that the actors are good but they cannot save the movie. A waste of time. The premise was not too bad. But one workable idea (interaction between real bussinessmen and Russian mafia) is not followed by an intelligent script **Revised Text:** We know from other movies that the actors are good and they make the movie. Not at all a waste of time. The premise was not bad. One workable idea (interaction between real bussiness men and Russian mafia) is followed by an intelligent script **Original Label:** Negative; **Revised Label:** Positive
SNLI	Substituting Entities	**Original Premise:** Several bikers are going down one side of a four lane road while passing buildings that seem to be composed mostly of shades of brown and peach. **Revised Premise:** Several bikers are going down one side of a four lane road while passing farms that seem to be composed mostly of shades of brown and peach. **Original Label:** Entailment; **Revised Label:** Contradiction

Table 1: Examples of few-shot categories from ANLI R3, IMDB, and SNLI datasets.

ing (French, 1999; Hassabis et al., 2017), defined as an adaptive system capable of learning from a continuous stream of information. The information progressively increases over time, but there is no predefined number of tasks to be learned. Majority of methods in continual learning focus on sequential training of various 'tasks' (not necessarily of same kind) and address the catastrophic forgetting problem. These approaches can be broadly classified into (1) architectural approaches that focus on altering the architecture of the network to reduce the interference between the tasks without changing the objective function (Razavian et al., 2014; Donahue et al., 2014; Yosinski et al., 2014; Rusu et al., 2016); (2) functional approaches that focus on penalizing the changes in the input-output function of the neural network (Jung et al., 2018; Li and Hoiem, 2017); and (3) structural approaches that introduce constraints on how much the parameters change when learning the new task so that they remain close to their starting point (Kirkpatrick et al., 2017). Other notable works in recent years are based on using intelligent synapses to accumulate task-related information over time (Zenke et al., 2017), using online variational inference (Nguyen et al., 2018), and dynamically expanding network capacity based on incoming data (Yoon et al., 2018). Further, a few previous works have explored continual learning with few examples for computer vision tasks (Le et al., 2019; Xie et al., 2019; Douillard et al., 2020; Tao et al., 2020). Unlike the typical continual learning setup, in our CFL, we continually learn various linguistic phenomena for the 'same task' with only limited labeled examples. Our setup is important for practical usage. The most closest to our work is from the vision community, where they proposed a benchmark suite containing few-shot datasets for continual learning and evaluation criteria (Antoniou et al., 2020). However, the major contrast is that our setup focuses on correcting the errors specific to a linguistic phenomenon rather than learning new class labels with few examples.

Online Learning. Online learning algorithms learn to update models from data streams sequentially, where the task is the same but can exhibit concept drift (new patterns) (Zinkevich, 2003; Crammer et al., 2006; Sahoo et al., 2018; Jerfel et al., 2019; Javed and White, 2019). Our setup is different from online learning because we start with a model that is fully trained on a task (i.e., no large sequential data steams), and only focus on correcting the errors specific to linguistic phenomena by giving few extra training examples.

3 Datasets

In this section, we describe all the English datasets that we curated and borrowed from previous works for creating a benchmark suite for continual few-shot learning (CFL). Table 1 presents some examples from these datasets.

3.1 ANLI R3 Few-Shot Categories

Nie et al. (2020) introduced the Adversarial Natural Language Inference (ANLI) dataset which consists of adversarially collected examples for Natural Language Inference (NLI) that are miss-classified by the current state-of-the-art models. The data is collected in three rounds with each round in-

	Numerical	Basic	Reference	Tricky	Reasoning	Imperfections
Train set	75 (15)	75 (15)	75 (15)	75 (15)	75 (15)	75 (15)
Dev set	49 (67)	154 (165)	76 (92)	70 (82)	205 (211)	32 (47)
Test set	117 (157)	360 (387)	179 (216)	164 (194)	481 (494)	77 (112)
Total	241 (239)	589 (567)	330 (323)	309 (291)	761 (720)	184 (174)

Table 2: Dataset statistics of 6 categories in ANLI R3 for few-shot learning setup (and continual few-shot setup).

troducing more difficult examples than the previous. Williams et al. (2020) analyzed the ANLI dataset and annotated the development set of all three rounds by labeling each example as to what type of reasoning is required to perform the inference. They used 40 fine-grained reasoning types organized hierarchically, where the top-level categories are: (1) *Numerical*: examples where numerical reasoning is crucial for determining the correct label. (2) *Basic*: require reasoning based on lexical hyponymy, conjunction, and negation. (3) *Reference*: noun or event references need to be resolved either within or between premise and hypothesis. (4) *Tricky*: require complex linguistic knowledge, e.g., pragmatics or syntactic verb argument structure. (5) *Reasoning*: require reasoning outside of the given premise and hypothesis pair. (6) *Imperfections*: examples that have spelling errors, foreign language content, or are ambiguous. We refer to Williams et al. (2020) for more details on each of these categories. We use the reasoning annotations to create a CFL setup. Unlike previous few-shot learning setups, we focus on few-shot learning of linguistic phenomena (6 categories in this case), instead of new tasks, classes, or domains.

We use the Round-3 (R3) development set and consider all 6 of the above categories as different few-shot learning cases (labeled ANLI R3 categories in the rest of the paper). In our framework, we consider two scenarios: (1) few-shot learning setup; (2) continual few-shot learning setup.[2] In the few-shot learning setup, for each category, we choose 5 disjoint training sets with each set containing 5 examples from each class label. The rest of the examples are divided into development and test sets with 30% and 70% splits, respectively. For each category in the continual few-shot learning setup, we choose 5 training examples from each class label. Training examples across the categories are disjoint, and we divide the rest of the examples in each category into development and test sets with 30% and 70% splits, respectively. Table 2 presents the full statistics on all 6 categories.

[2]The details of these setups are in Sec. 4.

3.2 SNLI Counterfactual Few-Shot Categories

Stanford NLI dataset (Bowman et al., 2015) is a popular natural language inference dataset where given a premise and a hypothesis, the task is to predict whether hypothesis entails or contradicts or neural w.r.t. the premise. Kaushik et al. (2020) annotated a small part of the SNLI dataset by modifying either the premise or hypothesis with minimum changes to create counterfactual target labels dubbed revised examples. We use the revised examples to create a few-shot learning setup. First, we train a RoBERTa-Large (Liu et al., 2019) classifier on the full original SNLI dataset which consists of ∼550K examples. We then filter examples from the revised data which are incorrectly predicted by the trained classifier. Then, we manually annotate these filtered examples based on the most frequent edit categories mentioned in Kaushik et al. (2020), along with some new additions. These categories are as follows (1) *Insert or remove phrases*: refers to examples where either phrases are added or removed in premise or hypothesis to change the label. (2) *Substitute entities*: refers to examples where changing the entity in premise or hypothesis enables to change the label. (3) *Substitute evidence*: refers to examples where changing the evidence in premise or hypothesis results in a change in the class label. (4) *Modify entity details*: refers to examples where the details of the entities are modified to change the label, e.g., red ball vs. blue ball. (5) *Change action*: refers to examples where a change in the action word in premise or hypothesis results in a change in the label, (6) *Numerical changes*: refers to change in numerical aspects results in a change in the label, e.g., one person vs. two persons. (7) *Negation*: refers to examples where negation is used to change the label. (8) *Using abstractions*: refers to examples where original words are replaced with their abstractions or vice-versa

Category	Train	Dev	Test	Total
IMDB				
Modifiers	30 (10)	42 (47)	99 (110)	171 (167)
Negation	30 (10)	19 (24)	45 (110)	95 (144)
SNLI				
Insert/remove phrases	45 (15)	141 (149)	331 (348)	517 (512)
Substitute entities	45 (15)	66 (74)	156 (174)	267 (263)
Substitute evidence	45 (15)	93 (100)	218 (236)	356 (351)
Modify entity details	45 (15)	36 (44)	85 (105)	166 (164)
Change action	45 (15)	26 (35)	63 (82)	134 (132)

Table 3: Dataset statistics of various categories in IMDB and SNLI counterfactual data for few-shot learning setup (and continual few-shot learning setup).

to change the label, e.g, man vs. person.[3] A few examples did not fall into any of these categories which are labeled as '*Other*', and are discarded. We follow similar data splits as discussed for ANLI R3 few-shot categories, except that we use only 3 training sets instead of 5 in the few-shot learning setup. We did not get enough balanced training sets for negation, numerical changes, and using abstraction categories, hence discarded them. The statistics of the rest of the categories are presented in Table 3.

3.3 IMDB Counterfactual Few-Shot Categories

Kaushik et al. (2020) also annotated a small part of the IMDB sentiment analysis dataset (Maas et al., 2011) by modifying the input examples with minimum changes to create counterfactual target labels. We follow a similar procedure as in Sec. 3.2 to create a few-shot learning setup from these revised examples. We categorize the examples as follows: (1) *Inserting or replacing modifiers*, (2) *Inserting phrases*, (3) *Adding negations*, (4) *Diminishing polarity via qualifiers*, (5) *Changing ratings*, and (6) *Suggesting sarcasm*. We discarded a few examples that did not belong to any of these categories. We follow similar data splits as discussed for SNLI counterfactual few-shot categories. We did not get enough balanced training sets for categories except inserting or replacing modifiers and adding negation, hence we discarded those categories. Table 3 presents the statistics of these two categories.

3.4 Annotation (More details in Appendix A)

First, a single expert annotated both SNLI and IMDB counterfactual examples, as both need a degree of expertise to correctly reason among various categories with examples often falling into multiple categories. Previous NLU projects also benefited from expert annotations (Basile et al., 2012; Bos et al., 2017; Warstadt et al., 2019; Williams et al., 2020). Next, since the annotations need complex reasoning and can be subjective sometimes, we further employed another annotator to annotate 100 examples from each dataset to calculate the inter-annotator agreement. We calculate the percentage agreement and Cohen's kappa (Cohen, 1960) for each category independently and report the average scores across all categories. The average percentage agreement score for SNLI and IMDB datasets are 86.4% and 90.5%, respectively, which is a high, acceptable level as per previous work (Toledo et al., 2012; Williams et al., 2020). The Cohen's kappa score (Cohen, 1960) for SNLI and IMDB datasets are 0.61 and 0.79, respectively, which is a *substantial* agreement (Landis and Koch, 1977).

4 Methods

Experimental Setup. In all experiments we first train a RoBERTa-Large (Liu et al., 2019) classifier on the original full training set (e.g., full SNLI data for SNLI few-shot categories). We then experiment with the curated few-shot datasets. We consider two setups: (a) few-shot learning, where we consider how methods adapt to a single error category; (b) continual few-shot learning setup, where methods 'continually' learn various error categories sequentially. The few-shot setup gives us an idea on how learnable is each error category/linguistic phenomenon with few examples, whereas the continual setting simulates a system that is repeatedly corrected. Next, we briefly discuss several baselines; more details on the baselines are in the Appendix.

Zero-Shot: Directly test the RoBERTa-Large classifier trained on the original data without using any few-shot training examples.

Fine-Tuning: Additionally fine-tune the original classifier with the few examples from the setup.

Memory-Aware Synapses: Aljundi et al. (2018) proposed an approach that estimates an importance weight for each parameter of the model, which approximates the sensitivity of the learned function to a parameter change. During the training with few-shot examples, the loss function is updated to consider the importance weights of the parameters through a regularizer.

[3] We refer to Sec. 3.4 for more details about the annotation.

Model	MNLI-m	MNLI-mm	Numerical	Basic	Reference	Tricky	Reasoning	Imperfections	Average
Zero-Shot	*90.3*	*90.1*	27.4	29.7	29.6	30.5	31.6	27.3	29.4
Fine-Tune	*90.2±0.1*	*90.0±0.1*	29.0±1.8	32.1±1.2	30.8±0.7	30.6±1.9	30.4±0.6	27.8±1.5	30.2±1.3
SCL	*90.3±0.0*	*90.0±0.1*	30.1±1.1	31.4±1.0	31.1±0.6	31.3±1.3	31.1±1.1	28.1±1.7	30.5±1.1
MAS	*90.2±0.1*	*89.9±0.1*	30.3±1.3	31.2±1.8	32.0±0.6	31.2±2.7	30.4±0.8	27.8±2.5	30.5±1.6
PN	*90.3*	*90.1*	37.6±8.6	42.7±6.4	44.0±7.0	37.7±4.6	43.0±6.7	38.2±5.6	40.5±6.5
k-NN	*89.8*	*89.7*	36.6±5.6	40.1±6.6	43.4±4.6	45.1±7.7	41.5±6.0	39.2±3.0	41.0±5.6

Table 4: Results on 6 categories of few-shot learning ANLI R3 dataset. Results are averaged across 5 support sets, and the corresponding standard deviation is also reported. Results reported in the last column ('Average') are based on the average performance on few-shot categories only.

Model	SNLI	Insert/Remove Phrases	Substitute Entities	Substitute Evidence	Change/Remove Entity Details	Change Action	Average
Zero-Shot	*92.5*	0.6	0.6	1.4	1.2	0.0	0.8
Fine-Tune	*92.3±0.2*	6.7±2.5	10.3±5.1	7.0±1.7	5.9±1.2	11.1±4.2	8.2±2.9
SCL	*92.3±0.3*	7.9±2.0	9.0±2.8	6.7±3.0	2.7±0.7	12.7±2.7	7.8±2.2
MAS	*92.1±0.2*	9.8±2.1	9.6±3.3	8.3±1.2	7.8±1.4	12.2±1.8	9.5±2.0
PN	*92.5*	48.6±7.5	44.2±9.2	47.9±1.5	44.3±9.5	40.7±14.4	45.1±8.4
k-NN	*92.3*	43.1±8.6	47.6±10.0	47.9±1.4	54.1±10.2	46.6±5.1	47.9±7.1

Table 5: Results on 5 categories of few-shot learning SNLI dataset. Results are averaged across 3 support sets.

Model	IMDB	Modifiers	Negation	Average
Zero-Shot	*96.0*	11.1	11.1	11.1
Fine-Tune	*96.0±0.0*	18.9±4.1	14.8±1.3	16.9±2.7
SCL	*96.0±0.0*	18.2±4.0	22.2±2.7	20.2±3.4
MAS	*95.9±0.0*	23.2±4.0	21.5±3.4	22.4±3.7
PN	*96.0*	88.9±1.0	89.6±1.3	89.3±1.2
k-NN	*95.8*	10.4±1.2	6.7±0.0	8.6±0.6

Table 6: Results on 2 categories of few-shot learning revised IMDB dataset. Results averaged across 3 sets.

Prototypical Networks (PN): Snell et al. (2017) proposed to produce a class distribution for an example based on a softmax over distances to the prototypes or mean class representations. In our work, we use several different support sets to compute the class prototypes: the original training data; the few-shot training examples; or, both. We use the output before softmax layer of the model (trained on cross-entropy loss using the original training data) as feature representation for examples (f_θ).

Supervised Contrastive Learning (SCL): Gunel et al. (2021) proposed supervised contrastive learning for better generalizability, where they jointly optimize the cross-entropy loss and supervised contrastive loss that captures the similarity between examples belonging to the same class while contrasting with examples from other classes.

k-Nearest Neighbors (k-NN): We recreate the classic nearest neighbors method by assigning to each example the dominant class label from the k-nearest training (support) examples. We measure the nearest examples based on the euclidean distance in the feature representation space f_θ.[4] We use the final encoder hidden representations before softmax layer as f_θ. As with Prototypical networks, support sets can be either the original training data, the few-shot training examples, or both.

5 Results

In this section, we report the performances of various baselines discussed in Sec. 4 on our benchmark suite. We refer to Appendix for training details.

5.1 Results on Few-Shot Learning

ANLI R3 Categories. Table 4 shows the results on the 6 categories from the Round-3 of the ANLI dataset. The base model, is trained on the combined data of MNLI (Williams et al., 2018), ANLI Round-1 (R1), and ANLI Round-2 (R2). On average, we observe that using the few-shot training examples for each of the categories improves the performance (comparing zero-shot vs. rest of the models), while maintaining the performance on MNLI matched (MNLI-m) and mis-matched (MNLI-mm) datasets. More importantly, we also observe that

[4]We use the faiss library (https://github.com/facebookresearch/faiss).

Model	MNLI	Numerical	Basic	Reference	Tricky	Reasoning	Imperfections	Average
Zero-Shot	90.3/90.0	33.1	29.5	32.4	28.9	32.2	28.6	30.8
Fine-Tune	90.0/89.7	34.4	35.1	**36.1**	**35.6**	35.2	30.4	34.5
SCL	89.9/89.6	32.5	**38.0**	**35.7**	**35.6**	**39.1**	30.4	35.2
MAS	90.4/90.1	32.5	31.3	31.9	30.9	33.2	25.9	31.0
Prototypical Network (PN)	83.7/83.2	28.0	**38.8**	33.8	27.8	**38.3**	**35.7**	33.7
Nearest Neighbor (k-NN)	89.9/89.7	31.8	30.7	30.1	29.9	31.8	25.9	30.0

Table 7: Continual learning results on few-shot ANLI R3 categories. Average score is on few-shot categories only. Bold numbers are statistically significantly better than the rest based on bootstrap test (Efron and Tibshirani, 1994).

Model	SNLI	Insert/Remove Phrases	Substitute Entities	Substitute Evidence	Change/Remove Entity Details	Change Action	Average
Zero-Shot	92.5	0.6	0.6	1.3	0.9	0.0	0.7
Fine-Tune	90.6	21.8	15.5	13.1	31.4	20.7	20.5
SCL	90.9	20.4	16.1	12.7	24.8	19.5	18.7
MAS	92.5	4.9	5.7	3.8	6.7	6.1	5.4
Prototypical Network (PN)	70.9	**44.3**	**40.8**	**44.1**	**36.2**	**52.4**	**43.6**
Nearest Neighbor (k-NN)	92.3	7.8	4.6	8.1	6.7	12.2	7.9

Table 8: Continual learning results on few-shot SNLI categories.

Model	IMDB	Modifiers	Negation	Average
Zero-Shot	96.0	11.1	11.1	11.1
Fine-Tune	96.0	30.9	33.9	32.4
SCL	95.2	26.4	26.8	26.6
MAS	96.1	14.5	10.7	12.6
PN	89.7	51.8	57.1	54.5
k-NN	96.0	10.9	8.9	9.9

Table 9: Continual learning results on few-shot IMDB categories.

simple feature matching-based approaches (Prototypical Networks (PN) and k-NNs) perform better than parameter correction approaches (e.g., fine-tuning, MAS, etc.) using the new examples as a support set. However, in the feature matching methods, we assume that we know whether the test example belongs to the original data or a linguistic category.[5, 6] Feature matching methods have higher variance than the parameter correction methods, as they are heavily dependent on the few-shot train examples (support set). However, feature matching methods still achieve remarkable performance with very few examples. We refer to Sec. 6 for more ablations on this interesting result. Note that the CFL problem is very challenging as it tries to correct the errors made by a well-trained model using only a few examples. Hence, many of our baselines have low scores on the categories. This further motivates the community in building new methods.

SNLI Categories. Table 5 presents the performance of various models on the 5 annotated categories of SNLI dataset in a few-shot learning setup. We observe similar trends: few-shot examples improve the performance (comparing zero-shot vs. other models in Table 5) and feature matching approaches perform consistently better than parameter correction approaches. Similar to the results on ANLI R3 categories, feature matching methods also exhibit high variance on the SNLI categories.

IMDB Categories. Table 6 presents the performance of various models on the 2 categories of few-shot IMDB sentiment analysis setup. Again, few examples improve the performance in all categories (with the exception of k-NN), and feature matching method (Prototypical Networks) outperforms parameter correction methods by a large margin. Since IMDB is a 2-way classification dataset and the examples are curated based on counterfactual edits, the feature matching methods have to figure out to just flip the label, which PN succeeded in (also reason for high scores) and k-NN did not in this case. Further, the variance for feature matching methods is notably lower on this dataset.

[5]The choice of a category as support set will provide the information on the test examples' category. In Table 10, we provide more results on avoiding this prior knowledge on the test examples' category.

[6]If we consider the categories as support set for calculating the scores on MNLI with PN, the performance drops from $90.3/90.1$ to $50.3 \pm 25.5/50.5 \pm 24.7$.

Model	MNLI	Numerical	Basic	Reference	Tricky	Reasoning	Imperfections
CE-Loss	90.3/90.1	27.4	29.7	29.6	30.5	31.6	27.3
PN with CE-Loss (†)	90.3/90.1	28.2	28.9	27.9	30.5	32.5	27.3
PN with CE-Loss (‡)	50.3±25.5/50.5±24.7	37.6±8.6	42.7±6.4	44.0±7.0	37.7±4.6	43.0±6.7	38.2±5.6
PN with CE-Loss (⋆)	87.6±0.8/87.2±1.2	31.8±4.3	35.3±1.7	34.4±5.1	31.7±3.5	34.0±4.7	28.6±7.3
PN with PN-Loss (†)	90.6/90.5	24.8	24.7	29.1	26.8	24.7	22.1
PN with PN-Loss (‡)	31.4±26.1/30.6±25.9	38.1±7.3	43.8±9.4	47.8±6.4	35.7±6.6	38.8±9.0	45.7±12.6
PN with PN-Loss (⋆)	90.4±0.3/90.2±0.3	27.0±2.2	27.4±1.9	32.5±2.9	27.8±1.4	27±3.4	26.5±3.4

Table 10: Comparison of various models with cross-entropy loss (CE-Loss) optimization or prototypical network loss (PN-Loss) optimization on NLI datasets. NLI models are trained with MNLI, ANLI R1, ANLI R2 data and tested on both matched/mismatched development sets of MNLI, and test sets of ANLI R3 categories. † represents MNLI as support set, ‡ represents ANLI R3 categories as support set, and ⋆ represents both as support set.

5.2 Results on Continual Few-Shot Learning

In this section, we discuss the continual few-shot learning setup on ANLI R3, SNLI, and IMDB categories. We sequentially train the models on each category by initializing with the model parameters learned for the previous category, thus enabling continual few-shot learning. Evaluation is performed on the final model that we get after continually training on all categories. Table 7, Table 8, and Table 9 present the continual few-shot learning results for our three category datasets. All the methods start with a RoBERTa-Large classifier trained on MNLI+ANLI-R1+ANLI-R2, full SNLI, and full IMDB datasets for their respective category datasets. Then, they are continually trained on each of the categories in the order as reported in the Tables. From the results, we observe that all methods perform better than the zero-shot method. Both fine-tuning and SCL approaches are doing better in this setup. PN has mixed results for ANLI R3 and good category-based results on SNLI and IMDB, but lowest test scores on the original datasets (MNLI, SNLI, and IMDB).[7,8] We hypothesize that equal weight of all class representations from the original dataset and categories leads to higher misclassification of test examples from the original dataset. The k-NN method has good results on the original dataset but lower scores on category-based results. Since the original dataset has more examples as support set than categories, we hypothesize that test examples from the categories could not effectively find relevant category-specific examples in their nearest neighbors.

[7]For the PN, we find the closest mean feature class from the pool of all mean feature classes of support sets that have so far appeared during the continual learning.

[8]For k-NN, we continually update the feature set with all the training examples that have so far appeared.

6 Ablations and Analyses

Robustness of Prototypical Networks. To ablate on how Prototypical networks (PN) performs on the original data (e.g., MNLI or SNLI or IMDB), we use the model trained with the cross-entropy loss and test it using PN with the original training data as the support set. Surprisingly, we observe that PN performs equal to that of general softmax-based prediction on all three datasets (see Table 10 row-1 vs. row-2, MNLI column; Table 11 row-1 vs. row-2, SNLI and IMDB columns). This is interesting since and we can simply calculate an example's Euclidean distance to the mean feature representations of classes to label it.

Cross-Entropy vs. Prototypical Loss. We train a model with Prototypical network (PN) loss (minimize the distance between training examples and the approximated class representations) and compare it with cross-entropy (CE) loss. Table 10 and Table 11 present the results. The model trained with PN loss performs similar or slightly better than cross-entropy loss on the original test sets (see Table 10 row-1 vs. row-5; Table 11 row-1 vs. row-5). Further, models with PN loss perform worse on average than the CE loss for ANLI R3 categories, whereas the opposite is true for the counterfactual categories of SNLI and IMDB (Table 11 row-1 vs. row-5). Note that ANLI R3 categories have examples from different domains that are not present in MNLI (Nie et al., 2020), whereas the categories of SNLI and IMDB have examples with counterfactual edits but same domain as the their full original datasets. This suggests that PN loss can generalize well to in-domain examples, but worse to out-of-domain examples.

We also tried combining both the original dataset

Model	SNLI	Insert/Remove Phrases	Substitute Entities	Substitute Evidence	Change/Remove Entity Details	Change Action	IMDB	Modifiers	Negation
CE-Loss	92.5	0.6	0.6	1.4	1.2	0	96.0	11.1	11.1
PN w/ CE-Loss (†)	92.5	2.1	1.9	2.8	4.7	3.2	96.0	12.1	11.1
PN w/ CE-Loss (‡)	8.0±0.1	48.6±7.5	44.2±9.2	47.9±1.5	44.3±9.5	40.7±14.4	4.0±0.0	88.9±1.0	89.6±1.3
PN w/ CE-Loss (⋆)	84.91±.7	17.8±12.5	23.5±6.5	29.1±1.3	11.8±7.1	23.3±8.7	88.1±2.1	60.9±10.4	57.0±5.1
PN w/ PN-Loss (†)	92.0	19.0	17.3	20.2	16.5	28.6	96.0	49.5	37.8
PN w/ PN-Loss (‡)	50.1±45.1	42.8±5.4	44.7±2.3	48.5±7.4	48.6±7.1	41.8±5.6	50.0±50.4	49.5±0.0	51.8±12.2
PN w/ PN-Loss (⋆)	90.2±0.6	19.0±0.0	17.3±0.0	20.2±0.0	16.5±0.0	28.6±0.0	95.9±0.1	49.5±0.0	41.5±3.4

Table 11: Comparison of the performance of various models with cross-entropy loss (CE-Loss) optimization or prototypical network loss (PN-Loss) optimization on NLI and sentiment analysis datasets. NLI models are trained on SNLI dataset and tested on test sets of SNLI and its counterfactual categories. IMDB models are trained on full IMDB dataset and tested on test sets of IMDB and its counterfactual categories. † represents SNLI/IMDB as support set, ‡ represents SNLI or IMDB categories as support set, and ⋆ represent both SNLI/IMDB and their categories as support set.

and the few-shot categories as the support set,[9] and observe a performance drop in the ANLI R3 few-shot categories, but still better than just using original dataset (MNLI) as support set (Table 10). This holds for both CE and PN losses. On the SNLI and IMDB categories setup, the performance drops again but still better than original dataset as support set on CE loss and almost same on PN loss.

t-SNE Plot Visualizations. To further understand the differences between cross-entropy loss and Prototypical network (PN) loss, we present t-SNE plots[10] on the examples from MNLI and ANLI R3 categories (each example is represented in the feature space f_θ). In Figure 1, the top row plots are based on a cross-entropy-trained NLI model (trained on MNLI, ANLI R1, and ANLI R2) and the bottom row based on PN-loss-trained NLI model. Each plot combines examples from MNLI and one of the ANLI R3 categories. It is evident that MNLI examples form class specific clusters. However, the ANLI R3 categories' examples may not belong to its label cluster of MNLI, suggesting their low performance in Table 4 zero-shot results. Interestingly most of these examples are at the edge of the clusters. Further, there is a remarkable difference in the cluster patterns between CE and PN loss models. CE loss plots have dense clusters and PN loss plots have skew (stretched) clusters. We also observe that clusters based on PN loss model have higher average distance to their cluster center and a higher average distance with very high variance between any two examples that belong to the same cluster, supporting the 2D t-SNE observations.

[9]For a given test example, we assign the class label of the closest mean class feature from the pool of mean class features of original train data and categories train data.

[10]sklearn library (https://scikit-learn.org/).

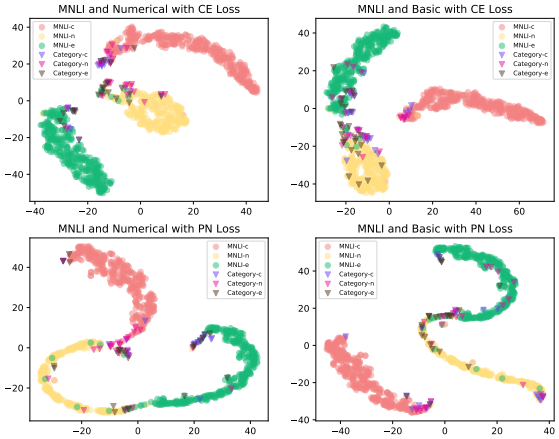

Figure 1: t-SNE plots showing examples from various classes.

7 Conclusion

We presented a benchmark suite and evaluation protocol for continual few-shot learning (CFL) on the text classification tasks. We presented several methods as important baselines for our CFL setup. Further, we provided several interesting ablations to understand the use of non-parametric feature matching methods for CFL. We hope that our work will serve as a useful starting point to encourage future work on this important practical setting.

Acknowledgments

We thank the anonymous reviewers for their helpful comments and Adina Williams for help with the ANLI R3 categories data. This work was supported by Facebook, DARPA YFA17-D17AP00022, ONR N00014-18-1-2871, and a Microsoft PhD Fellowship.

Broader Impact and Ethics Statement

We view the CFL as a way to make real-world AI systems safe and reliable by being able to correct errors quickly. At the same time, we believe there is a lot more to be done to bring the CFL approach to practical scenarios and we do not intend to directly employ our benchmark suite off-the-shelf on any real systems. Our benchmark suite serves only to compare various models and encourage the community to build better models on this important practical setting. Moreover, since CFL deals with only a few examples of training, the models might overfit these examples, so any practical usage of such setup should thoroughly consider the implications of overfitting scenarios. Further, our data collection methods for this research and the setup are not tuned for any specific real-world application. Hence, while applying our methods in a sensitive context, it is important to strictly employ extensive qualitative control and robust testing before using them with real systems.

References

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. 2018. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154.

Antreas Antoniou, Massimiliano Patacchiola, Mateusz Ochal, and Amos Storkey. 2020. Defining benchmarks for continual few-shot learning. *arXiv preprint arXiv:2004.11967*.

Valerio Basile, Johan Bos, Kilian Evang, and Noortje Venhuizen. 2012. Developing a large semantically annotated corpus. In *LREC 2012, Eighth International Conference on Language Resources and Evaluation*.

Johan Bos, Valerio Basile, Kilian Evang, Noortje J Venhuizen, and Johannes Bjerva. 2017. The groningen meaning bank. In *Handbook of linguistic annotation*, pages 463–496. Springer.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *NeurIPS*.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2014. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, pages 647–655.

Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. 2020. PODNet: Pooled outputs distillation for small-tasks incremental learning. In *Computer vision-ECCV 2020-16th European conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XX*, volume 12365, pages 86–102. Springer.

Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.

Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. 2021. Supervised contrastive learning for pre-trained language model fine-tuning. In *ICLR*.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809.

Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. 2017. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258.

Khurram Javed and Martha White. 2019. Meta-learning representations for continual learning. In *NeurIPS*.

Ghassen Jerfel, Erin Grant, Thomas L Griffiths, and Katherine Heller. 2019. Reconciling meta-learning and continual learning with online mixtures of tasks. In *NeurIPS*.

Heechul Jung, Jeongwoo Ju, Minju Jung, and Junmo Kim. 2018. Less-forgetting learning in deep neural networks. In *AAAI*.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *ICLR*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

Canyu Le, Xihan Wei, Biao Wang, Lei Zhang, and Zhonggui Chen. 2019. Learning continually from low-shot data stream. *arXiv preprint arXiv:1908.10223*.

Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Tsendsuren Munkhdalai and Hong Yu. 2017. Meta networks. *Proceedings of machine learning research*, 70:2554.

Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. 2018. Variational continual learning. In *ICLR*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *ACL*.

Sachin Ravi and Hugo Larochelle. 2017. Optimization as a model for few-shot learning. In *ICLR*.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. Cnn features off-the-shelf: an astounding baseline for recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 512–519. IEEE.

Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. 2018. Meta-learning for semi-supervised few-shot classification. In *ICLR*.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive neural networks. *arXiv preprint arXiv:1606.04671*.

Doyen Sahoo, Quang Pham, Jing Lu, and Steven CH Hoi. 2018. Online deep learning: Learning deep neural networks on the fly. In *IJCAI*.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze questions for few-shot text classification and natural language inference. In *EACL*.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208.

Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. 2020. Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12183–12192.

Assaf Toledo, Sophia Katrenko, Stavroula Alexandropoulou, Heidi Klockmann, Asher Stern, Ido Dagan, and Yoad Winter. 2012. Semantic annotation for textual entailment recognition. In *Mexican International Conference on Artificial Intelligence*, pages 12–25. Springer.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *NeurIPS*.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):1–34.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL*.

Adina Williams, Tristan Thrush, and Douwe Kiela. 2020. ANLIzing the adversarial natural language inference dataset. *arXiv preprint arXiv:2010.12729*.

Shudong Xie, Yiqun Li, Dongyun Lin, Tin Lay Nwe, and Sheng Dong. 2019. Meta module generation for fast few-shot incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0.

Wenpeng Yin, Nazneen Fatema Rajani, Dragomir Radev, Richard Socher, and Caiming Xiong. 2020. Universal natural language processing with limited annotations: Try few-shot textual entailment as a start. In *EMNLP*.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. 2018. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In *NeurIPS*, pages 3320–3328.

Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. 2018. Diverse few-shot text classification with multiple metrics. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1206–1215.

Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *ICML*, pages 3987–3995.

Richard Zens, Franz Josef Och, and Hermann Ney. 2002. Phrase-based statistical machine translation. In *Annual Conference on Artificial Intelligence*, pages 18–32. Springer.

Martin Zinkevich. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936.

A Annotation Details

Annotation of both SNLI and IMDB counterfactual examples needed a degree of expertise to correctly reason among various categories with often examples falling into multiple categories. Hence, a single expert manually annotated both datasets in an attempt to ensure high quality. The annotation process is not done at scale, so this approach seemed safer. ANLI categories discussed in Sec. 3.1 are also manually annotated by an expert (Williams et al., 2020). Further, various NLU projects benefited from expert annotations (Basile et al., 2012; Bos et al., 2017; Warstadt et al., 2019).

The expert annotated $1,422$ and 234 examples in SNLI and IMDB counterfactual datasets, respectively. It took roughly 15 hours to complete the annotations.

Inter-annotator Agreement. Since the annotations need complex reasoning and can be sometimes subjective, we further employed another annotator to annotate a subset of the examples to calculate the inter-annotator agreement. The new annotator first went over the definitions of various categories and later trained with a few examples. Finally, the new annotator annotated 100 examples each from SNLI and IMDB datasets.

We calculate the inter-annotator agreement on these second-annotated examples using the percentage agreement and Cohen's kappa (Cohen, 1960) for each category independently and report the average scores across all categories. For the SNLI counterfactual dataset, average percentage agreement score between the two annotators is 86.4%, and the average kappa score is 0.62. Our inter-annotator percentage agreement score is at an acceptable level as per previous work (Toledo et al., 2012; Williams et al., 2020) annotation agreement scores on similar types of annotations. Further, Cohen's kappa score ranges from -1 to 1, and a score in the range of 0.61 to 0.80 is considered as *substantial* agreement (Cohen, 1960; Landis and Koch, 1977). For the IMDB counterfactual dataset, the average percentage agreement score between the two annotators is 90.5, and the corresponding Cohen's kappa score is 0.79, which is again a *substantial* agreement.

B More Details on Baselines

Memory-Aware Synapses. Aljundi et al. (2018) proposed an approach that estimates an importance

weight for each parameter of the model which approximates the sensitivity of the learned function to a parameter change.

Let f be a function with parameters θ that represents the neural network model trained on the original full dataset. Let X, Y be the new examples from the few-shot setup. Hence, for a given data point x_k, the output of the network is $f(x_k; \theta)$. A small perturbation δ in the parameters space results in a change in the output function as follows:

$$f(x_k; \theta + \delta) - f(x_k; \theta) \approx \sum_{i,j} g_{ij}(x_k)\delta_{ij} \quad (1)$$

where $g_{ij}(x_k)$ is the gradient of the learned function w.r.t. the parameter θ_{ij} and δ_{ij} is the change in the parameter θ_{ij}. The magnitude of the gradient $g_{ij}(x_k)$ represents the importance of a parameter w.r.t. the input x_k, hence, the overall importance weight Ω_{ij} for a parameter θ_{ij} is defined as follows:

$$\Omega_{ij} = \frac{1}{N}\sum_{k=1}^{N} ||g_{ij}(x_k)|| \quad (2)$$

where N is the total number of few-shot examples. Aljundi et al. (2018) proposed to use l_2 norm of the function f to calculate g_{ij}, since this scalar value allows to estimate g_{ij} with a single back propagation. During the training with few-shot examples, the loss function is updated to consider the importance weights of the parameters through a regularizer. The final loss function is defined as follows:

$$L'(\theta) = L(\theta) + \lambda \sum_{i,j} \Omega_{ij}(\theta_{ij} - \theta^*_{ij})^2 \quad (3)$$

where λ is the hyperparameter for the regularizer and θ^*_{ij} is the learned parameter on the original full dataset.

Prototypical Networks. Snell et al. (2017) rely on an embedding function f_θ that computes an m dimensional representation for each example and a prototype for each class. Let X, Y represents a set of few-shot examples, then the class representation features are computed as $c_k = \frac{1}{|S_k|}\sum_{(x_k,y_k)\in S_k} f_\theta(x_k)$, where k represents the k^{th} class and S_k represents all the few-shot examples that belong to k^{th} class. Prototypical networks produce a class distribution for an example based on a softmax over distances to the prototypes or mean class representations (c_k). The class distribution for an example is defined as follows:

$$p_\theta(y = k|x) = \frac{\exp(-d(f_\theta(x), c_k))}{\sum_{k'} exp(-d(f_\theta(x), c_{k'}))} \quad (4)$$

where d is the Euclidean distance.

We use several different support sets to compute the class prototypes: the original training data; the few-shot training examples; or, both. We use the model's output before the softmax layer as $f_\theta(x)$. For our initial experiments, we use the model trained on cross-entropy loss using the original training data. We also experiment with a model trained on Prototypical loss (results discussion in Sec. 6), where we randomly sample a support set from the training data during each mini-batch optimization step and try to minimize the distance between the mini-batch examples and the approximated class representations based on the support set. Distance minimization is done using Eqn. 4.

Supervised Contrastive Learning (SCL). Gunel et al. (2021) proposed supervised contrastive learning for better generalizability, where they jointly optimize the cross-entropy loss and supervised contrastive loss that captures the similarity between examples belonging to same class while contrasting with examples from other classes. Let X, Y be the few-shot examples, then the total loss and supervised contrastive loss are defined as follows:

$$L = (1 - \lambda)L_{XE} + \lambda L_{SCL} \quad (5)$$

$$L_{SCL} = \sum_{i=1}^{N} -\frac{1}{N_{y_i} - 1} \sum_{x_j \in S_{y_i}} \log \frac{g_\theta(x_i, x_j)}{\sum_{k,i\neq k} g_\theta(x_i, x_k)} \quad (6)$$

$$g_\theta(x_i, x_j) = \exp(f_\theta(x_i) \cdot f_\theta(x_j)/\tau) \quad (7)$$

where λ is a hyperparameter to balance these two losses. τ is also a hyperparameter to control the smoothness of the distribution. N_{y_i} represent number of examples with class label y_i, and S_{y_i} represents the set of all the examples belonging to class label y_i. In this work, we use l_2 normalized representation of the final encoder hidden layer before softmax as f_θ.

C Original Datasets Details

In our experiments, before training on our category datasets, we initially train our RoBERTa-Large

Continual Training on (↓)	MNLI	Numerical	Basic	Reference	Tricky	Reasoning	Imperfections
MNLI+ANLI-R1+ANLI-R2	90.3/90.1	33.1	29.5	32.4	28.9	32.2	28.6
Numerical	90.3/90.0	33.1	30.0	32.9	29.9	33.2	27.7
Basic	90.2/89.9	31.8	31.3	32.4	32.5	32.6	29.5
Reference	90.3/89.9	32.5	33.9	33.3	35.6	33.4	32.1
Tricky	90.3/89.9	32.5	32.8	31.5	36.6	33.4	29.5
Reasoning	90.0/89.7	31.2	37.0	36.1	38.7	33.8	31.2
Imperfections	90.0/89.7	34.4	35.1	36.1	35.6	35.2	30.4

Table 12: Continual learning results on ANLI R3 categories using the fine-tuning method. Model is continually trained on each category in the order as presented in this Table, and tested on MNLI and all the categories of ANLI R3.

classifier on a base original dataset. For the ANLI R3 categories, we first train on MNLI, ANLI R1, and ANLI R2 training sets with $392,702$, $16,946$, and $45,460$ training examples, respectively.[11, 12] We report the performance on the development set of both matched and mis-matched examples of MNLI (Williams et al., 2018). The number of examples in the matched and mis-matched set are $9,815$ and $9,832$, respectively. Similarly, for the SNLI categories, we first train on the original SNLI (Bowman et al., 2015) with the number of examples in train, development, and test sets are $550,152$, $10,000$, and $10,000$, respectively.[13] For the IMDB categories, we use the data provided by Kaushik et al. (2020) with $19,262$ train examples and $20,000$ test examples.[14]

D Training Details

In all our experiments, we use the RoBERTa-Large classifier (356M parameters).[15] We report on accuracy for all of our models. Our choice of the best model during training is decided based on the accuracy performance on the development set. We do minimal manual hyperparameter search in our experiments. While training on the original datasets (MNLI+ANLI R1+ANLI R2, SNLI, or IMDB), we use a learning rate of $2e^{-5}$. For the training on the few-shot categories, we use a learning rate of $1e^{-5}$, where we initially tuned in the range [$2e{-}5$, $5e^{-6}$]. We keep the rest of the hyperparameters same between training on the original dataset versus training on the few-shot categories, e.g., we use a batch size of 32, maximum sequence length of 128 for training and 256 for testing, etc. The average run time for training on the few-shot categories is less than five minutes (because of very few training examples). We use 4 Nvidia GeForce GTX 1080 GPUs on a Ubuntu 16.04 system to train our models.

E Additional Results

E.1 Effect of Few-Shot Learning on Domains

In order to better understand the few-shot learning performance at the domain level, we chose the ANLI R3 few-shot learning setting where domain (genre) information is available. For example, the numerical category has 'Wikipedia' and 'RTE' domains. Table 13 presents the domain specific performances of various categories comparing parameter correction approach (fine-tuning) and non-parametric feature matching method (Prototypical Networks). We observe that 'Legal' domain performed best on average for both methods. Furthermore, the feature-matching method performed 'relatively' better on RTE domain whereas the parameter correction method performed relatively worse on this domain.

E.2 Fine-grained Continual Learning Results

Table 12 presents the detailed continual learning results on ANLI R3 categories using the fine-tuning method. First, we observe that the performance on MNLI drops as we add the categories, suggesting that it is affected by catastrophic forgetting. Next, we observe that the performance on all categories improve after the end of the continual training (w.r.t. performance on the pre-trained model). Further, we also observe that some categories are helping improve other categories. For example, after continually training the model from tricky category to reasoning category, the performance on

[11] https://gluebenchmark.com/tasks

[12] https://github.com/facebookresearch/anli

[13] https://nlp.stanford.edu/projects/snli/

[14] https://github.com/acmi-lab/counterfactually-augmented-data

[15] Based on Transformers repository (https://github.com/huggingface/transformers).

	Fine-Tune	PN
Numerical		
- Wikipedia	33.2±5.4	36.3±8.0
- RTE	28.6±1.9	38.7±10.9
Basic		
- Legal	34.9±1.9	46.7±7.8
- Procedural	32.2±3.6	43.5±4.3
- Wikipedia	34.3±2.6	39.0±6.9
- RTE	28.4±1.7	41.7±10.6
Reference		
- Legal	35.7±2.5	48.3±11.1
- Wikipedia	31.5±2.0	38.8±5.4
- RTE	26.9±1.1	46.3±9.3
Tricky		
- Legal	27.1±2.4	40.0±3.8
- Procedural	36.4±6.4	45.5±6.4
- Wikipedia	33.7±2.1	38.8±7.5
- RTE	30.2±1.5	33.3±6.4
Reasoning		
- Legal	34.0±1.9	42.6±7.0
- Procedural	31.2±2.2	44.2±11.0
- Wikipedia	31.1±1.4	38.6±4.8
- RTE	25.7±1.8	47.5±12.1
Imperfections		
- Legal	66.7±0.0	46.7±18.3
- Wikipedia	36.5±1.6	34.7±7.0
- RTE	17.5±3.1	40.5±13.0

Table 13: Performance of parameter correction approach (fine-tuning) and non-parametric feature matching method (Prototypical Networks - PN) on various domains of ANLI R3 few-shot categories.

the basic category drastically improved, suggesting that reasoning category has some useful information to improve the performance on basic category. Similarly, performance on the reference category also improved dramatically, suggest that reasoning examples are useful for learning reference linguistic phenomenon. This suggests that ordering of these categories has influence on the performance to a certain degree. In order to understand this impact, we randomly selected 10 different orders and performed the continual learning of ANLI R3 categories. We observed (1) a standard deviation of 1.0 on the average scores; (2) the performance of all categories except reasoning are relatively more sensitive to the ordering.