Label Semantic Aware Pre-training for Few-shot Text Classification

Aaron Mueller[1*] **Jason Krone**[2] **Salvatore Romeo**[2]
Saab Mansour[2] **Elman Mansimov**[2] **Yi Zhang**[2] **Dan Roth**[3,2]
[1]Department of Computer Science, Johns Hopkins University
[2]Amazon Web Services AI Labs
[3]Department of Computer & Information Science, University of Pennsylvania
amueller@jhu.edu, {kronej,romeos,saabm,mansimov,yizhngn,drot}@amazon.com

Abstract

In text classification tasks, useful information is encoded in the label names. Label semantic aware systems have leveraged this information for improved text classification performance during fine-tuning and prediction. However, use of label-semantics during pre-training has not been extensively explored. We therefore propose **L**abel **S**emantic **A**ware **P**re-training (LSAP) to improve the generalization and data efficiency of text classification systems. LSAP incorporates label semantics into pre-trained generative models (T5 in our case) by performing secondary pre-training on labeled sentences from a variety of domains. As domain-general pre-training requires large amounts of data, we develop a filtering and labeling pipeline to automatically create sentence-label pairs from unlabeled text. We perform experiments on intent (ATIS, Snips, TOPv2) and topic classification (AG News, Yahoo! Answers). LSAP obtains significant accuracy improvements over state-of-the-art models for few-shot text classification while maintaining performance comparable to state of the art in high-resource settings.

1 Introduction

Large pre-trained language models have enabled better performance on many NLP tasks—especially in few-shot settings (Brown et al., 2020; Schick and Schütze, 2021a; Wu and Dredze, 2020). More informative representations of textual inputs often leads to much higher downstream performance on NLP applications, which explains the rapid and general adoption of models such as (Ro)BERT(a) (Devlin et al., 2019; Liu et al., 2019), GPT-2 (Radford et al., 2019), and T5 (Raffel et al., 2020). However, while these models are often used to effectively encode inputs, fewer works have attempted to give models access to informative representations of *labels* as well.

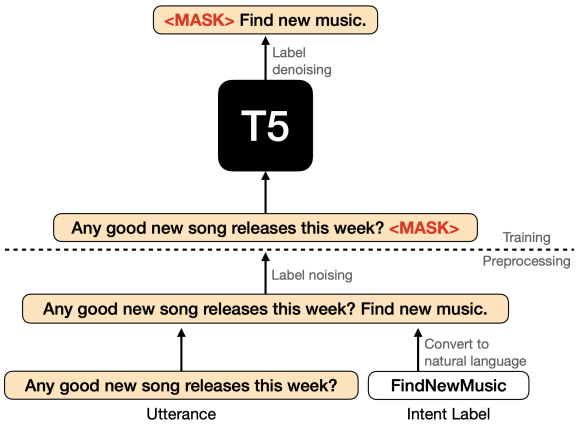

Figure 1: Overview of our approach, label semantic aware pre-training (LSAP). We collect utterance-intent pairs and create new pairs from unlabeled Reddit and Twitter data, convert the intents to natural language, concatenate the utterance and intent, noise the concatenated sequence, and train a sequence-to-sequence model to denoise the sequence.

Most discriminative approaches to text classification only give the model access to label indices. A recent stream of work has obtained significant improvements in structured prediction tasks by using sequence-to-sequence (seq2seq) models to generate labels (Athiwaratkun et al., 2020; Paolini et al., 2021). Yet these generative approaches make use of label semantics—the meaning of class label names—only during fine-tuning and prediction. Thus, we propose **L**abel **S**emantic **A**ware **P**re-training (LSAP) to incorporate label semantics as well as input-label associations into the pre-training step (Figure 1). Our experiments show that LSAP yields higher performance with fewer fine-tuning examples in a variety of domains.

Our contributions include the following:

1. A method to incorporate label semantics into generative models during pre-training.
2. A method for creating utterance-intent pairs for label semantic aware pre-training from unlabeled noisy data.

*Work done as an intern at Amazon Web Services.

3. State-of-the-art few-shot performance on intent and topic classification datasets.

Our code is publicly available.[1]

2 Related Work

Label semantics has been leveraged in many settings and tasks to improve performance and robustness, even before dense embedding representations became standard: Chang et al. (2008) achieve over 80% accuracy on binary text classification tasks without any labeled training examples by giving naïve Bayes classifiers rich semantic representations (Gabrilovich and Markovitch, 2007) of labels. Song and Roth (2014) use a similar approach for hierarchical and multinomial classification, finding that dataless procedures approached the performance of (and sometimes outperformed) supervised approaches.

More recently, label semantics based on dense embedding representations have become popular—especially with the rise of contextualized word embeddings (Peters et al., 2018). One stream of work integrates label representations using label vectors and label attention: label-wise attention networks (LWAN; Mullenbach et al., 2018) are designed for datasets with very large structured label spaces, and where multiple labels can apply to a potentially very long document. Mullenbach et al. (2018) generate label feature vectors and use an LWAN based on convolutional neural networks (CNNs); Rios and Kavuluru (2018) extend the attention mechanism for zero-shot settings. Chalkidis et al. (2020) use BERT to embed the labels for an LWAN.[2] In contrast, our label space is flatter and much smaller, and the input texts are much shorter on average.

Another label-semantic-aware approach for massively multi-label text classification is CLESS (Rethmeier and Augenstein, 2022). CLESS performs contrastive pre-training from scratch on a question-answering dataset with a large and sparse label space. Using contrasts between positive and negative answer embeddings, they obtain performance comparable to or better than RoBERTa using only in-domain data. However, CLESS pretrains and classifies on the same data, and is meant for classification in one domain with a very large label space; our setting contains a larger mismatch between pre-training (where we have domain-general data and a large label space) and fine-tuning (where we have domain-specific data and a small label space). We want our approach to be more domain-general and to leverage the simplicity and effectiveness of token and span reconstruction objectives—for example, masked language modeling (Devlin et al., 2019) and span denoising (Raffel et al., 2020). Thus, we opt to perform secondary pre-training on data from a variety of domains using an existing model.

Other work integrates label embeddings into short-text intent and topic classification systems, more similarly to our task. Gaonkar et al. (2020) use label embeddings from BERT and a label attention mechanism to improve emotion classification accuracy. Generative approaches like those of Rongali et al. (2020); Athiwaratkun et al. (2020); Paolini et al. (2021) implicitly make use of label semantics for text and token classification tasks by generating the labels at prediction time. Rastogi et al. (2019) use embeddings of human-defined schema which guide a dialogue state tracking system.

Few-shot text classification entails performing classification after training or tuning a model on only a few examples from the training split of an evaluation set. Recent approaches to this include (Ro)BERT(a)-based (Chen et al., 2020) and especially XLNet-based (Yang et al., 2019) classifiers, prototypical networks (Snell et al., 2017), dynamic memory induction networks (Geng et al., 2020), and generative classification (discussion follows).

Text classification contains more specific subtasks. Here, we evaluate on topic classification (TC; labeling text with its general domain, e.g. "world news") and intent classification (IC; labeling intentful text with what it is trying to accomplish, e.g. "book flight"). The scarcity of IC data in many domains prevents the use of many neural text classification methods (Krone et al., 2020). In IC, the input is a conversational utterance and the output is a label describing what the user intends to do; for example, given a set of intents {`BookHotel`, `BookFlight`, `CheckAccount`} and an input utterance "I would like a flight to NYC next month", the model should classify the utterance as `BookFlight`. Recent approaches to few-shot IC include dual encoders (Cer et al., 2018; Henderson et al., 2020;

[1] https://github.com/amazon-research/label-aware-pretrain

[2] Note that BERT-Base often performs similarly to or better than BERT-LWAN, and XLNet performs significantly better than BERT-Base in text classification settings that more closely resemble our setting (Yang et al., 2019). We therefore opt to compare to XLNet.

Casanueva et al., 2020), combining prototypical networks with meta-learning (Krone et al., 2020), a nearest-neighbor discriminative method (Zhang et al., 2020a), and span-level contextualized embedding retrieval (Yu et al., 2021).

Generative text classification has become more feasible with the advent of large pre-trained language models (Radford et al., 2019; Brown et al., 2020). Here, one tunes a model to generate a natural language label given an input sequence, which reduces the train-test gap and does not require architectural changes.

Pattern-exploiting training (PET; Schick and Schütze, 2021a,b) entails formatting train and test examples as cloze-style prompts, where the label is generally one word. Here, a language model can see the inputs and the embeddings of the output classes during tuning and prediction, though unlike our approach, there is no domain-general training step. Prompt tuning (Lester et al., 2021; Gao et al., 2021) improves PET by automatically optimizing the prompt design. Our approach is even more widely applicable in that we give the model concatenated utterances and labels and allow it to transduce to variable-length label sequences, rather than reformatting the evaluation data into a cloze format where the label must be one word/token.

Sequence-to-sequence (seq2seq) approaches based on pointer/copy mechanisms (Rongali et al., 2020) or T5 (Raffel et al., 2020) tend to be effective and more data-efficient for text and token classification in semantic parsing tasks, including slot labeling (Athiwaratkun et al., 2020) and entity-relation extraction (Paolini et al., 2021). In this method, one trains or fine-tunes a seq2seq model to transduce from an unlabeled text sequence to a sequence with labeled spans (or to just the label). Our approach is based on T5, but unlike prior work, we perform a *label semantic aware* secondary pre-training step on a variety of datasets before fine-tuning.

3 Approach

Our approach, LSAP, performs a secondary pre-training step with T5 on a large scale collection of (pseudo-)labeled examples; see Figure 1 for an overview. In addition to training on existing labeled datasets (§3.1), we (1) filter unlabeled data using a **dialogue act classifier** (§3.1.1); (2) pseudo-label the utterances that pass the filter using an **intent generator** (§3.1.1); and (3) perform secondary pre-training on the labeled and pseudo-labeled data using T5 (§3.2). §3.1 details how we collect, select, and label pre-training data. §3.2 describes the pre-training formats we test in our experiments.

3.1 Data

Our pre-training data (Table 1) consists of utterances with intent labels. We use intentful utterances for pre-training because intent labels are often more specific, informative, and varied than sentiment or topic labels. For example, consider the intents `BookFlight` and `ViewAirfare` versus the topic `AirTravel`. Our pre-training corpus combines gold (human-labeled), silver (heuristically labeled), and bronze (pseudo-labeled) data.

We first collect gold datasets: here, we use a set of non-public benchmark datasets as well as PolyAI Banking (Coope et al., 2020), which yields approximately 130K training examples containing over 1,200 unique intents.[3]

To supplement this data, we add the silver WikiHow intents (Zhang et al., 2020b) dataset, which is heuristically labeled. Each WikiHow example consists of an utterance, the longest step in a WikiHow article, and an intent label (the article title with "How To" removed). Training on this dataset has been shown to improve few-shot IC performance in a variety of domains (Zhang et al., 2020b).

3.1.1 Pseudo-labeling Noisy Data

Gold and silver conversational datasets are scarce and tend to focus on one or a few narrow domains. To obtain more data from a larger variety of domains, we propose a filtering and labeling pipeline for converting unlabeled conversational data into pseudo-labeled "bronze" pre-training data. See Figure 2 for an overview.

We start by collecting conversational utterances. We first use the Customer Support on Twitter (CSTwitter) dataset;[4] this consists of over 2.81M tweets to and from company customer support agents. We also use the Reddit PushShift dataset;[5] this dataset is large, so we only download most recent comment dumps by date until reaching 100M comments.

Dialogue act classifier. In a conversational dataset, many utterances will not be intentful, and thus will not lend themselves to informative labels. For example, the statement "Hiking is an

[3] We ensure that there is no overlap between the gold data and our evaluation sets.

[4] https://www.kaggle.com/thoughtvector/customer-support-on-twitter

[5] https://files.pushshift.io/reddit/comments/

Dataset	Quality	Description	Training Examples		Intents
Internal benchmark data	Gold	Benchmark datasets. 15+ domains from English locales.		119,920	1,172
PolyAI Banking	Gold	Online banking queries.		10,016	77
WikiHow Intent Classification	Silver	Automatically labeled intent classification dataset. Intent is article title with "How to" removed, and utterance is the longest step in the article.		110,573	110,573
			Pre-filter	**Post-filter**	
Customer Support on Twitter	Bronze	Conversations (tweets) between consumers and customer support agents on Twitter.	2,811,774	446,309	122,909
Reddit PushShift	Bronze	Comments scraped from Reddit.	100,000,000	680,000	220,786

Table 1: Pre-training datasets. "Quality" refers to the source of the labels (human-labeled is gold, deterministically labeled is silver, probabilistically labeled is bronze). "Pre-filter" and "Post-filter" refer to the number of training examples before and after using the dialogue act classifier described in §3.1.1.

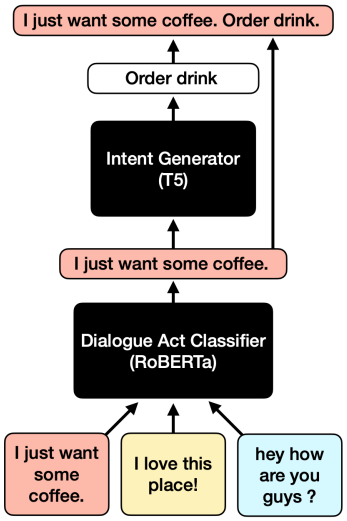

Figure 2: Our pipeline for creating utterance-intent pairs from unlabeled conversational data. We automatically filter data for intentful utterances using a dialogue act classifier, and then run filtered utterances through a T5-based intent generator. Each filtered utterance and its respective intent is concatenated to create a pre-training example.

outdoor activity" does not express a clear goal out-of-context (i.e., it is non-intentful), whereas "Buy a plane ticket to NYC for next month" is a command with a clear goal. Applying an intent label to a non-intentful utterance may lead to supervision that is harmful to downstream performance. To filter for intentful utterances, we tune a RoBERTa-based binary classifier (intentful vs. non-intentful) using Multi-Domain Goal-Oriented Dialogue (MultiDoGO; Peskov et al., 2019) and Schema-guided Dialogue (SGD; Rastogi et al., 2019; Kale and Rastogi, 2020). For MultiDoGO, we treat greetings/goodbyes, thank yous, and other generic intents as non-intentful/negative examples; we treat all other intents that are not out-of-domain as intentful/positive examples. For SGD, we treat any utterance tagged with INFORM intents as non-intentful; utterances with intents tagged as REQUEST are treated as intentful. When evaluating on a held-out set of MultiDoGO and SGD, the classifier achieves 98% precision.

To evaluate the precision of the classifier on our newly filtered data, we randomly sample 150 utterances (per dataset) tagged as intentful by the classifier and calculate the proportion that are actually intentful, as judged by human evaluation. For CSTwitter, we obtain 91% precision; this high precision may be due to the dataset being composed primarily of intentful customer-service-focused queries. For Reddit, we initially obtained 54% precision. We qualitatively find that the probability assigned by the classifier to the positive label correlates well with intentfulness, so for Reddit, we exclude all examples to which our classifier assigned a positive-label probability lower than the median for all utterances tagged as positive examples. After probability thresholding, we obtain 76% precision on Reddit.

Intent generator. To label the intentful utterances, we train a T5-based generative intent labeler. We fine-tune T5 on the gold and silver data to transduce from utterances to intents (e.g., "intent classification: Find me a hotel in NYC" → "Book hotel"), and then apply this tuned model to the filtered utterances. We find that this model gen-

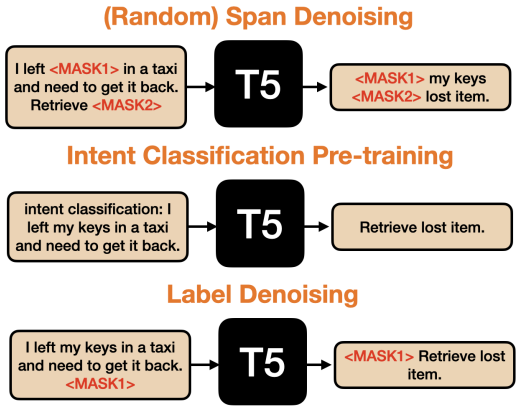

Figure 3: Pre-training formats that we test in our experiments. We find that label denoising is best.

erates intents not seen in the training set for 37% of the utterances. These novel intents often but do not always demonstrate lexical overlap with the utterance (e.g., "i'm wondering if there's fireworks for sale there?" $\rightarrow$ "Fireworks for sale") or add specific descriptors to intents from the training set (e.g., "Find a job" appears in the training set, but our model generates "Find a job in the UK" which does not).

3.2 Pre-training Formats

We experiment with different pre-training formats, all but one of which are based on T5's span denoising objective. See Figure 3 for an overview of our formats and an example of each.

First, we try *random span denoising*. In this approach, we concatenate each intent label (in natural language format) to its associated utterance. Then, we randomly noise 15% of the tokens[6] in the utterance-intent input sequence, reconstructing the contiguous noised spans in the output sequence. This is the same objective that T5 uses.[7]

Our second approach is *intent classification (IC) pre-training*, where we supervise T5 with utterances and intents in the same format used for downstream supervised fine-tuning. Here, the input sequence is "intent classification: " followed by the utterance. The output sequence is the intent.

Finally, we implement *label denoising*, where we train on utterances and their respective intents as before; however, instead of noising random spans, we deterministically noise the entire label sequence in the input sequence and reconstruct it in the output sequence. This is a way of framing the intent classification task as an unsupervised denoising task. This format is formally equivalent to IC pre-training in that we must transduce from utterances to intents, though we empirically show that the unsupervised denoising format matters a great deal for downstream IC performance.

3.3 Evaluation Label Overlap is Rare

We search for non-case-sensitive intent matches with Snips and ATIS in our pre-training dataset, finding that exact intent matches (or intents with Snips/ATIS intents as substrings) are very rare in our data: 682 examples (0.005%) featured exact or substring overlap. We also search for the presence of *any* lexical overlap between the intents in ATIS/Snips and the intents in our pre-training examples (i.e., if any word in an ATIS or Snips intent appears in a pre-training example's intent, we count it): less than 8,000 examples (0.6%) in our pre-training dataset featured lexical overlap. Utterances tagged with exact intent matches were often not similar to the kinds of utterances seen in the evaluation sets; for example, `PlayMusic` is a label in Snips that typically refers to playing specific songs or artists, while the same label in our pre-training set often referred to requests to buy or play an instrument.

To check whether there was semantic overlap more broadly between the utterances in our pre-training set and those in Snips/ATIS, we obtain sentence embeddings for 5 randomly sampled utterances from each intent in Snips and ATIS, and for each utterance in our pre-training dataset. Sentence embeddings are obtained using a sentence-BERT model (Reimers and Gurevych, 2019); we use `all-MiniLM-L6-v2`, as it is both fast and performs best on semantic similarity tasks. We then calculate pairwise semantic similarity between the utterances in our pre-training set and the Snips/ATIS utterances by calculating the cosine similarity between sentence embedding pairs. When observing the most similar utterances, we do not often see much semantic overlap. For example, "Play some Mf Doom from the sixties on pandora" (in Snips) has the greatest sentence embeddings similarity to "I love doom and won't let some launcher protest stop me from playing it" (in our dataset); there is

[6] 15% is the proportion of tokens that are noised in the original pre-training setup (Raffel et al., 2020).

[7] Unlike T5, we do not pack multiple sequences into single training examples. When we pack multiple utterance-intent pairs into training examples of length $\approx$ 512, we find that performance drops sharply compared to training on individual utterance-intent pairs.

lexical overlap in the utterance, but the sentence in our dataset refers to playing a video game rather than a song. However, there were rare instances where semantic overlap was significant: for example, "Add this tune to the Rock Save the Queen playlist" (in Snips) was similar to "Please add the Fresh Prince of Bel-Air song to this" in our dataset.

4 Experimental Setup

We aim to understand whether LSAP can improve text classification performance over a variety of state-of-the-art (SOTA) baselines.

Dataset	Train	Test	Classes	Balanced?
Snips	13,784	700	7	✓
ATIS	4,978	700	20	
TOPv2 (reminder)	494	338	9	
TOPv2 (weather)	177	148	4	
Yahoo! Answers	1.4M	60,000	10	✓
AG News	120,000	7,600	4	✓

Table 2: Evaluation datasets for intent classification (top) and topic classification (bottom). For TOPv2, we use the two provided low-resource domains as separate evaluation sets; we use the 25 Samples per Intent and Slot (SPIS) splits from Chen et al. (2020) as the maximum split size.

We evaluate LSAP on two text classification tasks: intent classification (IC) and topic classification (TC). IC is the most similar task to our pre-training objective. Our IC evaluation datasets include Snips (Coucke et al., 2018), ATIS (Price, 1990), and the low-resource *reminder* and *weather* domains provided in TOPv2 (Chen et al., 2020). Snips' intents are each from different domains. ATIS focuses on the flight domain; some intents appear only once or not at all in the training set, and some utterances are tagged with multiple intents (we count these as separate intents). For multi-class examples, we separate intents with the character "#" (e.g., "book flight # airfare"). TOPv2 (*reminder*) focuses on the creation and modification of personal reminders, while TOPv2 (*weather*) focuses on queries regarding the weather and time of sunrise/sunset.

For TC, we evaluate on Yahoo! Answers (YA)[8] and AG News.[9] YA consists of conversational user questions labeled with the general category of the question (e.g., "Health", "Sports"). AG News consists of formal-register news articles labeled with their category (e.g., "Business", "World").

4.1 Fine-tuning

When fine-tuning our T5-based models, the input sequence consists of an "intent classification:" prefix followed by the utterance. The output sequence is the intent in a natural language format. For example: "intent classification: Find me a flight from NYC to Baltimore." → "Book flight".[10]

We create few-shot splits of size k for each evaluation set, where we sample $\leq k$ examples for each class i from the set of labeled training examples T. More specifically, $\forall k \in \{1, 2, 4, 8, 16, 32\}$,$\forall i \in T$, if $|T_i| \geq k$, we randomly sample k training examples from T_i *without replacement*; if $|T_i| < k$, we simply use T_i. We ensure that smaller splits are subsets of larger splits such that we may perform more principled comparisons across split sizes. In low-resource settings, the random seed can have a large impact on performance. We therefore average all accuracies over 5 random fine-tuning seeds.

See Appendix G for hyperparameters.

4.2 Baselines

XLNet (Yang et al., 2019) is an autoregressive Transformer-based (Vaswani et al., 2017) language model which sees various word order permutations of the inputs during pre-training. We append a linear layer to the mean-pooled output of XLNet to obtain an XLNet-based classifier. This model has achieved SOTA performance on many text classification datasets, including AG News.

SEQ2SEQ-PTR (Rongali et al., 2020) is based on sequence-to-sequence (Sutskever et al., 2014) and pointer generator networks (Vinyals et al., 2015; See et al., 2017); it achieves SOTA IC/SL performance on Snips, ATIS, and TOPv2. It transduces from an unlabeled utterance to a sequence with labeled spans (including the intent label, where the span is the entire sentence).

LM-BFF (Gao et al., 2021) is a prompt-tuning-based model for few-shot text classification. It uses automatically generated and optimized prompts (from T5) to perform generative classification (with RoBERTa) by predicting a masked token after the

[8] https://www.kaggle.com/soumikrakshit/yahoo-answers-dataset

[9] http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html

[10] We also try label denoising fine-tuning to reduce the mismatch between pre-training and fine-tuning, but we find that this is not as effective as the traditional T5 fine-tuning setup. See Appendix F for scores.

	Snips			ATIS			TOPv2 (reminder)			TOPv2 (weather)		
Format	1-shot	4-shot	Full	1-shot	4-shot	Full	1-shot	4-shot	Full	1-shot	4-shot	Full
Span denoising	80.9	92.0	**99.1**	67.0	85.4	**97.8**	60.9	80.0	**92.0**	61.5	77.1	86.4
IC pre-training	67.1	90.9	99.0	67.0	86.9	97.4	68.2	77.4	87.1	70.5	80.1	87.1
Label denoising	**88.7**	**93.5**	99.0	**68.7**	**87.8**	97.6	**69.0**	**80.6**	91.4	**72.7**	**81.4**	**89.1**

Table 3: 1-shot, 4-shot, and full-resource accuracies across LSAP pre-training formats on each IC evaluation set. Label denoising is consistently best in few-shot settings while obtaining similar performance to other formats in full-resource settings. Subsequent tables refer to the label denoising variant as LSAP for brevity. Results for all models and split sizes in Appendix C.

input. It achieves SOTA scores on few-shot binary/trinary sentiment classification and NLI tasks. We employ the best-performing *prompt-tuning with demonstrations* approach, adapting it for IC by reducing the intent labels to single words.

T5 is the closest baseline to LSAP: our model differs only in the addition of the label aware pre-training step. We thus compare LSAP against T5 to understand the contribution of our label aware pre-training approach to downstream performance. T5 is comparable to TANL (Paolini et al., 2021), which is based on vanilla T5; TANL's SOTA performance on a variety of structured prediction tasks indicates that T5 as-is can be a strong baseline.

To test whether performance improvements with LSAP can be attributed to label semantics or domain adaptation on the utterances, we also present results for T5 adapted (with random span denoising) on only on the utterances and *not* the intents in our pre-training data. This is equivalent to LSAP with random span denoising, but without the intent labels. We refer to this as "T5 (adapted)".

5 Results

Which pre-training format is best? We present 1-shot, 4-shot, and full-resource IC accuracies on each evaluation set in Table 3. **Label denoising is the best-performing pre-training format in few-shot settings across all evaluation sets**. Differences in performance between LSAP formats decrease as we increase the fine-tuning set size; in full-resource settings, the difference between span denoising and label denoising is not significant. This suggests that explicitly demarcating intents from utterances during pre-training may help T5 better leverage the pre-training examples.[11] As "label denoising" performs best, we focus on that variant of LSAP from here on, though note that all LSAP formats outperform T5 and T5 (adapted).

Table 4 displays IC accuracies across models. Compared to T5, the T5 (adapted) baseline achieves consistently higher IC accuracies. However, T5 (adapted) is vastly outperformed by LSAP (by up to 18% on Snips and 3% on ATIS in 1-shot settings), even though the utterances are the same across these settings; this indicates that **much of the improvements may be attributed to label semantics**, and that domain adaptation is only responsible for a small portion of the improvement.

	Snips: **Examples per Label**						
Model	1	2	4	8	16	32	Full
XLNet	70.0	77.6	88.1	92.9	96.2	96.9	99.0
LM-BFF	75.3	82.2	88.3	94.0	96.5	**97.6**	98.9
SEQ2SEQ-PTR	75.8	84.2	89.5	93.5	96.2	97.1	99.0
T5	71.1	79.5	89.5	92.9	95.2	96.5	99.0
T5 (adapted)	74.9	81.3	91.2	94.4	96.2	96.9	98.9
LSAP	**88.7**	**90.5**	**93.5**	**94.8**	**96.7**	97.3	99.0
	ATIS: **Examples per Label**						
Model	1	2	4	8	16	32	Full
XLNet	24.1	46.8	70.2	77.2	92.4	94.4	**98.0**
SEQ2SEQ-PTR	15.6	31.6	45.8	77.0	83.3	95.3	97.4
T5	45.8	78.7	83.9	90.3	92.3	94.5	97.4
T5 (adapted)	66.9	78.0	84.5	91.7	93.6	95.5	97.6
LSAP	**68.7**	**79.5**	**87.8**	**92.4**	**95.7**	**96.4**	97.6

Table 4: Mean intent classification accuracies across 5 seeds on Snips and ATIS at various few-shot split sizes. Smaller splits are subsets of larger splits. LSAP is consistently best in lower-resource settings while maintaining comparable performance to other models in higher-resource settings. Standard deviations and results for TOPv2 are in Appendix C.

We next observe that **generative approaches are generally more effective than discriminative approaches, especially in lower-resource settings.** The XLNet classifier does not have access to label semantics during fine-tuning, instead observing class indices and utterances only; all other approaches hence have access to more information during fine-tuning and prediction.

[11] These results are stable across random samples of the few-shot splits. See Appendix B.

	Yahoo! Answers: **Examples per Label**						
Model	1	2	4	8	16	32	Full
XLNet	23.9	42.4	44.0	52.8	62.3	65.9	77.6
T5	41.9	54.0	59.3	61.3	64.6	65.6	**77.8**
LSAP	**49.2**	**58.8**	**60.7**	**63.3**	**64.7**	**66.4**	77.7
	AG News: **Examples per Label**						
Model	1	2	4	8	16	32	Full
XLNet	55.3	63.5	69.7	79.4	**85.4**	**86.8**	**95.6**
T5	62.0	74.4	77.2	81.1	84.6	85.9	94.8
LSAP	**74.8**	**77.2**	**80.7**	**82.3**	**85.4**	86.5	94.8

Table 5: Few-shot text classification accuracies on Yahoo! Answers (top) and AG News (bottom). Smaller splits are subsets of larger splits. Our LSAP approach yields improvements over the SOTA XLNet model. LSAP also improves up over vanilla T5, indicating that our approach generalizes to topic classification.

	Snips: **Examples per Label**					
Pre-training Data	1	2	4	8	16	32
None	71.2	79.5	89.5	92.9	95.2	96.5
gold	82.2	86.9	91.7	94.0	95.9	96.5
+silver	81.6	85.3	92.0	95.0	96.0	96.8
+CSTwitter	82.9	88.4	92.7	94.4	**96.7**	97.1
+Reddit	**88.7**	**90.5**	93.5	94.8	**96.7**	97.3
CSTwitter+Reddit	80.9	90.1	**93.7**	**95.2**	96.6	**97.6**
C4	82.4	87.1	92.8	94.9	96.2	97.3

Table 6: Pre-training dataset ablation using LSAP (with label denoising). We display intent classification accuracies on Snips. Each pre-training dataset improves performance, but we can recover most of the performance from our best dataset when using *only* our bronze data.

LSAP induces better performance than LM-BFF and SEQ2SEQ-PTR in very low-resource settings, though LM-BFF and SEQ2SEQ-PTR perform better than T5. Note that for ATIS, LM-BFF does not effectively handle the multi-class examples due to the single-word cloze format, nor the high number of similar labels due to the intended contrastive word use case; we thus do not compare to LM-BFF here.

As we increase the size of the fine-tuning splits, performance increases and converges across methods. Thus, the primary contribution of LSAP is inducing quicker generalization across domains.

Finally, we present topic classification accuracies for YA and AG News (Table 5). Our LSAP approach yields up to 35% improvements on AG News over the SOTA XLNet model in 1-shot settings, and over 100% on YA; we also maintain comparable performance to the SOTA XLNet baseline in full-resource settings. LSAP also improves up to 21% on AG News and 18% on YA over vanilla T5; this is evidence that **our pre-training procedure is not just tuning T5 to recognize utterance-intent associations: it is also teaching T5 how to be a better text classifier in general.** This also indicates that our procedure generalizes to labels that have not been seen during pre-training: topic labels are not present in the pre-training data, though some of these labels do appear as substrings of more specific intent labels (e.g., while "Health" does not appear as a label in our pre-training data, "Get health information" does appear).

Our model also generalizes well to joint IC/SL. See Appendix A.

5.1 Dataset Ablation

Using LSAP with label denoising, we ablate over each pre-training dataset and observe the effect on downstream IC performance (Table 6).

The gold data improves few-shot IC performance over T5 by over 10%. Adding the silver data does not significantly change performance. Adding CSTwitter improves performance in few-shot settings, though it also increases variance across random seeds. The best accuracies and lowest variances are achieved after adding the Reddit data.

Notably, we find that **pre-training on only automatically filtered and labeled examples still improves performance over T5.** To test whether this is due to CSTwitter and Reddit simply being well-suited to our evaluation sets, we try automatically filtering and labeling a less conversational dataset: Colossal Cleaned Common Crawl (C4; Raffel et al., 2020), the same dataset used for pre-training T5. We use the same number of training examples as in our combined CSTwitter+Reddit data for comparability (1,126,309 examples). We still observe performance improvements when pre-training on this dataset, and the difference between pre-training on this versus CSTwitter+Reddit is only significant at 2 examples per label. This is evidence that **our method for creating new pre-training examples is effective with different types of data, including non-conversational data**. Note that adding C4 to the gold+silver+CSTwitter+Reddit data does not significantly improve performance; we therefore do not include it in our final pre-training dataset.

			Snips: **Examples per Label**					
Model	Shuffled pre-train labels?	Remapped eval labels?	1	2	4	8	16	32
T5	N/A	✓	-18.0	-9.9	-0.9	-0.4	-1.4	-0.2
T5 (adapted)	N/A	✓	-21.6	-11.9	-7.8	-2.9	+0.1	-0.1
LSAP	✗	✓	-23.6	-9.4	-1.8	-1.7	0.0	-0.2
LSAP	✓	✗	-42.4	-27.0	-15.7	-14.2	-17.4	-14.1
LSAP	✓	✓	-40.6	-31.9	-17.9	-10.6	-15.4	-8.9
			ATIS: **Examples per Label**					
Model	Shuffled pre-train labels?	Remapped eval labels?	1	2	4	8	16	32
T5	N/A	✓	-17.2	-25.2	-7.0	-6.2	-9.0	-5.8
T5 (adapted)	N/A	✓	-41.1	-29.7	-9.7	-7.9	-4.3	-1.3
LSAP	✗	✓	-35.7	-29.6	-10.6	-6.9	-4.5	-2.8
LSAP	✓	✗	-35.4	-19.5	-17.6	-15.4	-8.8	-9.0
LSAP	✓	✓	-41.8	-30.4	-36.0	-28.0	-21.7	-16.7

Table 7: Absolute difference in intent classification accuracy relative to LSAP on Snips and ATIS at all few-shot split sizes after shuffling labels assigned to utterances in the pre-training set (random label semantics), remapping labels in the evaluation set (misleading label semantics), or both.

5.2 The Importance of Meaningful Labels

Including label sequences in the pre-training data results in large performance increases. Is this due solely to intent sequences being helpful independent information, or is the model learning useful associations between the inputs and labels? To investigate this, we experiment with misleading and random label semantics. For "misleading label semantics," we remap the labels in an evaluation dataset by defining a bijective function from each intent to a randomly selected different intent, ensuring that identity mappings do not occur. In other words, if label i is replaced with label j, all instances of i are systematically converted to j in the training *and* test sets. We then fine-tune our models on these remapped-intent datasets and observe whether performance decreases, and by how much. Significant performance decreases indicate reliance on input-label associations.

We also define a "random label semantics" variant of LSAP, where we randomly shuffle the intents with respect to the utterances in the *pre-training data* and then pre-train with label denoising on the shuffled data. Unlike the evaluation label remapping, this shuffling procedure preserves the number of instances of each intent such that our pre-training set is technically the same—the intents and utterances are simply mismatched.

Performance with misleading and/or random label semantics (Table 7)[12] decreases considerably across datasets, and this is especially apparent in the low-resource case. Label denoising seems the most sensitive to utterance-label associations. Performance drops decrease with increasing few-shot split sizes. This suggests that **our models do rely on utterance-intent associations to achieve high IC performance, and that these associations are more important in lower-resource settings than higher-resource settings.**

[12]Misleading label semantics results for TOPv2 are in Appendix D.

6 Conclusions

We have proposed a pre-training approach for leveraging the semantic information inherent in labels. Our method improves few-shot text classification performance across domains while maintaining high performance in full-resource settings. This approach is fairly general and could potentially be extended to more structured semantic parsing tasks by annotating some of the pre-training examples, or perhaps by simply including diverse labeled examples from a variety of tasks in the pre-training step. Future work could investigate extending this method to pre-training from scratch, as well as tuning the utterance and label formats. One could also use demonstrations to achieve better performance with fewer gradient updates.

Acknowledgments

We thank Giovanni Paolini, Ben Athiwaratkun, and the reviewers for their thoughtful feedback on earlier versions of this work.

References

Ben Athiwaratkun, Cicero Nogueira dos Santos, Jason Krone, and Bing Xiang. 2020. Augmented natural language for generative sequence labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 375–385, Online. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *Computing Research Repository*, arXiv:1803.11175.

Ilias Chalkidis, Manos Fergadiotis, Sotiris Kotitsas, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. An empirical study on large-scale multi-label text classification including few and zero-shot labels. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7503–7515, Online. Association for Computational Linguistics.

Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 2, pages 830–835, Chicago, Illinois. Association for the Advancement of Artificial Intelligence.

Xilun Chen, Asish Ghoshal, Yashar Mehdad, Luke Zettlemoyer, and Sonal Gupta. 2020. Low-resource domain adaptation for compositional task-oriented semantic parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5090–5100, Online. Association for Computational Linguistics.

Samuel Coope, Tyler Farghly, Daniela Gerz, Ivan Vulić, and Matthew Henderson. 2020. Span-ConveRT: Few-shot span extraction for dialog with pretrained conversational representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 107–121, Online. Association for Computational Linguistics.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *Computing Research Repository*, arXiv:1805.10190.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of the Twentieth International Joint Conference on AI (IJCAI)*, pages 1606–1611.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Radhika Gaonkar, Heeyoung Kwon, Mohaddeseh Bastan, Niranjan Balasubramanian, and Nathanael Chambers. 2020. Modeling label semantics for predicting emotional reactions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4687–4692, Online. Association for Computational Linguistics.

Ruiying Geng, Binhua Li, Yongbin Li, Jian Sun, and Xiaodan Zhu. 2020. Dynamic memory induction networks for few-shot text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1087–1094, Online. Association for Computational Linguistics.

Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. 2020. ConveRT: Efficient and accurate conversational representations from transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2161–2174, Online. Association for Computational Linguistics.

Mihir Kale and Abhinav Rastogi. 2020. Template guided text generation for task oriented dialogue. In

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6505–6520, Online. Association for Computational Linguistics.

Jason Krone, Yi Zhang, and Mona Diab. 2020. Learning to classify intents and slot labels given a handful of examples. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 96–108, Online. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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. *Computing Research Repository*, arXiv:1907.11692.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. 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 1101–1111, New Orleans, Louisiana. Association for Computational Linguistics.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *International Conference on Learning Representations*.

Denis Peskov, Nancy Clarke, Jason Krone, Brigi Fodor, Yi Zhang, Adel Youssef, and Mona Diab. 2019. Multi-domain goal-oriented dialogues (MultiDoGO): Strategies toward curating and annotating large scale dialogue data. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4526–4536, Hong Kong, China. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Patti Price. 1990. Evaluation of spoken language systems: The ATIS domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2019. Towards Scalable Multi-domain Conversational Agents: The Schema-Guided Dialogue Dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Honolulu, Hawaii. Association for the Advancement of Artificial Intelligence.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Nils Rethmeier and Isabelle Augenstein. 2022. Long-tail zero and few-shot learning via contrastive pretraining on and for small data. In *Proceedings of the Workshop on Artificial Intelligence with Biased or Scarce Data (AIBSD) at the 36th AAAI Conference on Artificial Intelligence*, Online. Association for the Advancement of Artificial Intelligence.

Anthony Rios and Ramakanth Kavuluru. 2018. Few-shot and zero-shot multi-label learning for structured label spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3132–3142, Brussels, Belgium. Association for Computational Linguistics.

Subendhu Rongali, Luca Soldaini, Emilio Monti, and Wael Hamza. 2020. Don't parse, generate! A sequence to sequence architecture for task-oriented semantic parsing. In *Proceedings of The Web Conference 2020*, page 2962–2968, New York, New York. Association for Computing Machinery.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also

few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30, Long Beach, California. Curran Associates, Inc.

Yangqiu Song and Dan Roth. 2014. On dataless hierarchical text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, Québec City, Canada. Association for the Advancement of Artificial Intelligence.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27, Montréal, Canada. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, Long Beach, California. Curran Associates, Inc.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, volume 28, Montréal, Canada. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32, Online. Curran Associates, Inc.

Dian Yu, Luheng He, Yuan Zhang, Xinya Du, Panupong Pasupat, and Qi Li. 2021. Few-shot intent classification and slot filling with retrieved examples. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 734–749, Online. Association for Computational Linguistics.

Jianguo Zhang, Kazuma Hashimoto, Wenhao Liu, Chien-Sheng Wu, Yao Wan, Philip Yu, Richard Socher, and Caiming Xiong. 2020a. Discriminative nearest neighbor few-shot intent detection by transferring natural language inference. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5064–5082, Online. Association for Computational Linguistics.

Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020b. Intent detection with WikiHow. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 328–333, Suzhou, China. Association for Computational Linguistics.

A Joint Intent Classification and Slot Labeling

Intent classification (IC) and slot labeling (SL) are often performed simultaneously. For sequence-to-sequence models, this can be done by transducing from an unlabeled utterance to a labeled utterance, as in Athiwaratkun et al. (2020). To evaluate whether our model can be used as an effective IC/SL system, we use the same approach, substituting our model for TANL. This tuning setup is more distinct from our pre-training task than the IC setup, so we hypothesize that performance gains over TANL will be slightly smaller here on IC than in the approach in the main paper. We also hypothesize that we will see similar SL scores as for TANL.

Our results (Table 8) indicate that in this setting, our pre-training approach still results in increased intent classification accuracy over TANL. As found before, the performance improvements are most noticeable in the few-shot setting. We also find that slot labeling performance is mostly maintained after pre-training, including in the 1-shot setting. Future work could investigate ways to integrate slot labeling supervision into pre-training for improving performance on both IC and SL.

		Intent Class.		Slot Labeling	
	Model	**ATIS**	**Snips**	**ATIS**	**Snips**
Full	Joint BERT	98.6	97.5	**97.0**	**96.1**
	ELMO+BiLSTM	**99.3**	97.4	93.9	95.6
	TANL	99.0	97.0	96.9	**96.1**
	LSAP (label denoising)	99.1	**97.6**	96.8	**96.1**
1-shot	TANL	78.8	88.5	**36.0**	**81.7**
	LSAP (label denoising)	**82.3**	**89.3**	35.8	80.8

Table 8: Intent classification accuracy and slot labeling F1 on ATIS and Snips.

B Stability Across Random Samples of the Few-shot Splits

In few-shot settings, the selection of fine-tuning examples can make a large difference in downstream performance. Here, we present IC accuracy distributions across 5 random samples of the 1-shot results (Figure 4) to understand whether the relative performance of each approach is stable given different fine-tuning set samples. Specifically, we present macroaverages across random fine-tuning set samples after averaging performance across random seeds; this is so that we average over more stable estimates of performance within each fine-tuning set.

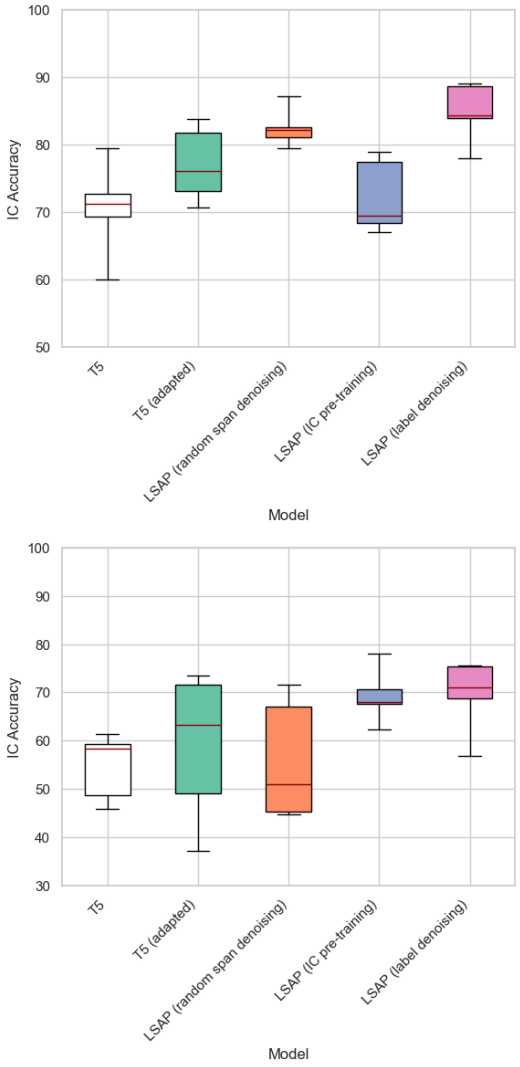

Figure 4: Intent classification accuracy distribution across 5 random samples of the 1-shot fine-tuning set for Snips (top) and ATIS (bottom).

Performance on Snips across random 1-shot fine-tuning samples is largely stable. All of our original conclusions regarding which models are better than others still hold: label denoising is best among our label semantic aware methods, and each label semantic aware approach (except IC pre-training) beats the non-label semantic aware baselines. Performance on ATIS is more variable across random 1-shot samples, but the relative performance of models is still largely the same: IC pre-training is more competitive on this evaluation set and random span denoising does not beat the T5 (adapted) baseline, but label denoising is still best.

C Pre-training Method Comparison: Full Results

Here, we present intent classification accuracies for ATIS (Table 10) the low-resource domains of TOPv2 (Tables 11,12) using each of our pre-training approaches. Our results here indicate that the relative performance of each method is stable across evaluation sets: **label denoising is consistently the best approach in the lowest-resource setting.** Intent classification pre-training is often second-best, while random span denoising is consistently least effective among our pre-training formats (but still improves performance over vanilla T5 and the T5 (adapted) baseline for *all* evaluation datasets).

D The Importance of Meaningful Labels: Full Results

Here, we present intent classification performance with misleading label semantics (after the random label remapping procedure described in §5.2) for the low-resource domains of TOPv2 (Tables 13,14). We present accuracy *differences* after remapping labels in order to observe how performance changes when using intent names that are not semantically related to the utterances they classify.

We find that our results are mostly consistent across evaluation sets: our best models rely more on utterance-intent associations and are thus more sensitive to label remapping, as evidenced by higher performance drops after label remapping. However, there appears to be variance across datasets with respect to reliance on intent labels: the performance drops for TOPv2 in both domains are much larger than for ATIS and Snips, providing evidence that **utterance-label associative information is much more important for some datasets than others**. We also find larger performance drops for our model pre-trained with the IC pre-training format than for our best model pre-trained with label denoising. Thus, **the best model is not necessarily the most reliant on utterance-label associations**.

The "random label semantics" baselines are sensitive to misleading label semantics for TOPv2 (reminder) and ATIS, as indicated by consistently large accuracy drops after label remapping. This is not the case for TOPv2 (weather) nor Snips in low-resource settings, where performance differences tend to be positive or closer to 0. The performance drop is much smaller for this baseline than for other

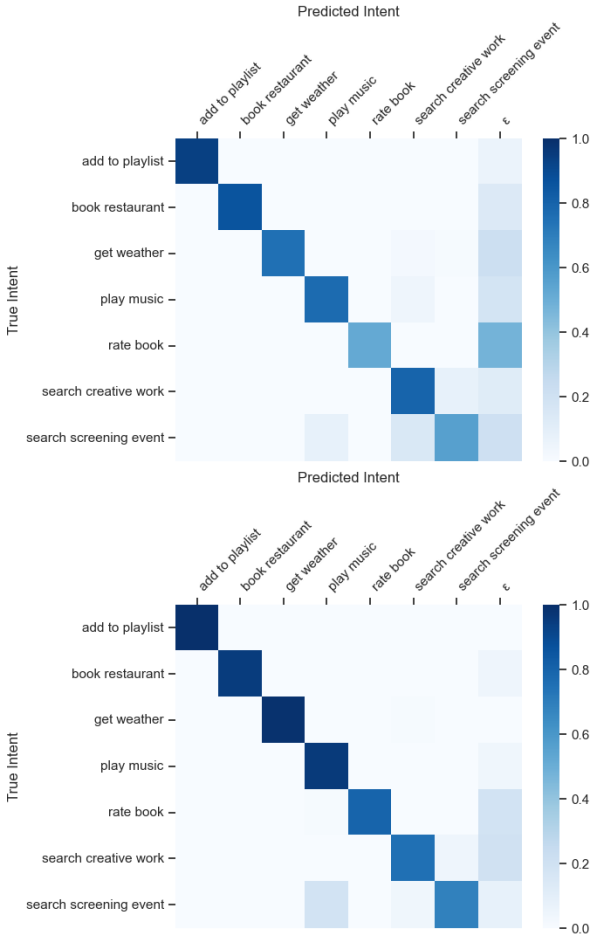

Figure 5: Confusion matrices displaying 1-shot results on Snips using vanilla T5 (top) and LSAP with label denoising (bottom). The fine-tuning setups are identical. "ϵ" refers to generated labels outside the set of labels in Snips.

models (indicating lower associative sensitivity), but it is still notable that the model was still able to rely on associations between utterances and intents after shuffling; perhaps this is due to the similarity of each intent label in these highly domain-specific datasets, though this would not explain the lack of sensitivity to utterance-label associations in TOPv2 (weather) since this is also highly domain-specific. Future work could more specifically investigate the source of these label sensitivity differences across datasets.

E Error Analysis

What kinds of errors does T5 make before and after pre-training? We present confusion matrices before and after our best label semantic aware pre-training approach (Figure 5). When using the same fine-tuning setup and hyperparameters, vanilla T5 tends to generate more out-of-domain intent labels

	Snips: **Examples per Label**						
Model	1	2	4	8	16	32	Full
XLNet	70.0	77.6	88.1	92.9	96.2	96.9	99.0
LM-BFF	75.3 (3.0)	82.2 (3.6)	88.3 (3.4)	94.0 (0.5)	96.5 (0.5)	**97.6** (0.4)	98.9
SEQ2SEQ-PTR	75.8 (2.1)	84.2 (2.1)	89.5 (0.9)	93.5 (1.0)	96.2 (0.3)	97.1 (0.4)	99.0
T5	71.1 (2.2)	79.5 (2.8)	89.5 (1.1)	92.9 (1.7)	95.2 (0.7)	96.5 (0.6)	99.0
T5 (adapted)	74.9 (4.7)	81.3 (3.1)	91.2 (1.0)	94.4 (0.5)	96.2 (0.3)	96.9 (0.3)	98.9
LSAP							
Span denoising	80.9 (2.0)	85.7 (1.6)	92.0 (1.3)	94.3 (1.2)	96.5 (0.2)	97.1 (0.7)	**99.1**
IC pre-training	67.1 (2.1)	83.2 (1.4)	90.9 (0.8)	94.5 (0.5)	96.5 (0.4)	97.4 (0.3)	99.0
Label denoising	**88.7** (1.5)	**90.5** (0.9)	**93.5** (0.8)	**94.8** (0.7)	**96.7** (0.4)	97.3 (0.3)	99.0

Table 9: Pre-training method comparison. Mean intent classification accuracies across 5 seeds on Snips at various few-shot split sizes. Smaller splits are subsets of larger splits. LSAP is consistently best in lower-resource settings while maintaining comparable performance to other models in higher-resource settings.

	ATIS: **Examples per Label**						
Model	1	2	4	8	16	32	Full
XLNet	24.1	46.8	70.2	77.2	92.4	94.4	**98.0**
SEQ2SEQ-PTR	15.6 (5.7)	31.6 (4.0)	45.8 (7.1)	77.0 (5.2)	83.3 (2.1)	95.3 (0.7)	97.4
T5	45.8 (18.2)	78.7 (7.1)	83.9 (2.9)	90.3 (1.3)	92.3 (1.7)	94.5 (1.0)	97.4
T5 (adapted)	66.9 (4.4)	78.0 (6.1)	84.5 (2.6)	91.7 (1.0)	93.6 (1.6)	95.5 (0.8)	97.6
LSAP							
Span denoising	67.0 (6.0)	77.6 (6.2)	85.4 (3.1)	91.4 (1.4)	94.6 (1.3)	95.5 (0.9)	97.8
IC pre-training	67.0 (3.0)	77.0 (1.1)	86.9 (2.1)	90.9 (1.0)	94.9 (0.8)	95.5 (0.8)	97.4
Label denoising	**68.7** (14.3)	**79.5** (8.1)	**87.8** (2.5)	**92.4** (7.9)	**95.7** (0.7)	**96.4** (0.8)	97.6

Table 10: Pre-training method comparison. Intent classification accuracy on ATIS at various few-shot split sizes. Smaller splits are subsets of larger splits. Each score is averaged over 5 fine-tuning runs. Note that the high standard deviation for T5 (label denoising) in the 1-shot setting is because the accuracies skew high; three seeds yield accuracies over 80%, one at 76.9%, and one at 60%.

than our best model. These out-of-domain generations are typically mergers between two intents: Snips contains `PlayMusic` and `RateBook` intents, and vanilla T5 often generates `RateMusic`. After pre-training, this is much less frequent.

Vanilla T5 also seems to assign higher prior probabilities to specific intents, despite seeing class-balanced tuning data; for example, `SearchCreativeWork` is generated for utterances from a variety of intents, as indicated by the slight vertical stripe in Figure 5. After pre-training, this problem is almost non-existent.

F Label Denoising Fine-tuning

Here, we compare the performance of LSAP (label denoising pre-training) when fine-tuned using the traditional T5 format (task prefix, no masking) and when using the label denoising format (concatenate the document and its label in the source sequence, mask the label sequence, and reconstruct the label in the output sequence). See Tables 15 and 16.

In lower-resource settings, regular fine-tuning significantly outperforms label denoising fine-tuning across evaluation sets; an exception is TOPv2 (*weather*) at 2 examples per label, but the difference in performance is not large here. In higher-resource settings, label denoising fine-tuning begins to achieve comparable performance to regular fine-tuning (and sometimes outperforms regular fine-tuning). Nonetheless, accuracy differences in higher-resource settings are not large, and regular fine-tuning performs significantly better on average. We therefore opt to use regular fine-tuning when comparing to baselines.

	TOPv2 (reminder): **Examples per Label**				
Model	1	2	4	8	25SPIS
XLNet	33.5	39.8	51.9	76.0	89.0
T5	51.9 (5.3)	61.6 (6.3)	72.3 (2.5)	83.3 (3.0)	91.7
T5 (adapted)	58.5 (3.6)	66.6 (2.3)	73.4 (4.0)	85.2 (2.6)	90.8
LSAP					
Span denoising	60.9 (5.0)	68.4 (4.0)	80.0 (1.6)	**88.6** (1.6)	92.0
IC pre-training	68.2 (3.1)	67.8 (1.8)	77.4 (1.5)	86.2 (1.3)	87.1
Label denoising	**69.0** (4.2)	**71.3** (3.8)	**80.6** (2.6)	87.7 (0.5)	91.4

Table 11: Pre-training method comparison. Intent classification accuracy on TOPv2 (*reminder* domain) at various few-shot split sizes. Smaller splits are subsets of larger splits. Each score is averaged over 5 fine-tuning runs.

	TOPv2 (weather): **Examples per Label**				
Model	1	2	4	8	25SPIS
XLNet	44.9	54.4	68.7	79.7	87.8
T5	53.2 (6.1)	66.5 (11.8)	74.4 (5.8)	83.1 (0.5)	87.1
T5 (adapted)	61.2 (3.8)	72.0 (4.5)	77.4 (1.8)	84.9 (1.3)	83.7
LSAP					
Span denoising	61.5 (8.1)	73.2 (3.0)	77.1 (3.3)	83.7 (0.7)	86.4
IC pre-training	70.5 (3.6)	**77.4** (1.5)	80.1 (1.0)	**83.8** (0.5)	87.1
Label denoising	**72.7** (1.3)	**77.4** (2.0)	**81.4** (1.1)	83.4 (0.8)	**89.1**

Table 12: Pre-training method comparison. Intent classification accuracy on TOPv2 (*weather* domain) at various few-shot split sizes. Smaller splits are subsets of larger splits. Each score is averaged over 5 fine-tuning runs.

			TOPv2 (reminder): **Examples per Label**			
Model	Shuffled pre-train labels?	Remapped eval labels?	1	2	4	8
T5	N/A	✓	-13.3	-16.1	-25.8	-21.6
T5 (adapted)	N/A	✓	-25.2	-26.8	-26.4	-17.2
LSAP	✗	✓	-38.6	-34.7	-28.2	-16.9
LSAP	✓	✗	-44.5	-29.9	-30.2	-20.0
LSAP	✓	✓	-53.4	-40.3	-46.5	-23.2

Table 13: Absolute difference in intent classification accuracy relative to LSAP on TOPv2 (*reminder* domain) at all few-shot split sizes after shuffling labels assigned to utterances in the pre-training set, remapping labels in the evaluation set, or both.

			TOPv2 (weather): **Examples per Label**			
Model	Shuffled pre-train labels?	Remapped eval labels?	1	2	4	8
T5	N/A	✓	-7.2	-20.9	-1.4	-10.1
T5 (adapted)	N/A	✓	-19.5	-28.7	-17.0	-13.6
LSAP	✗	✓	-34.3	-37.3	-19.7	-8.7
LSAP	✓	✗	-43.4	-37.1	-17.2	-17.0
LSAP	✓	✓	-37.1	-34.7	-22.7	-18.4

Table 14: Absolute difference in intent classification accuracy relative to LSAP on TOPv2 (*weather* domain) at all few-shot split sizes after shuffling labels assigned to utterances in the pre-training set, remapping labels in the evaluation set, or both.

	Snips: **Examples per Label**					
Model	1	2	4	8	16	32
LSAP						
FT	**88.7**	**90.5**	**93.5**	**94.8**	96.7	97.3
LDFT	75.2	84.9	92.0	93.7	**96.9**	**97.6**
	ATIS: **Examples per Label**					
LSAP						
FT	**68.7**	**79.5**	**87.8**	92.4	**95.7**	**96.4**
LDFT	59.7	76.3	85.7	**93.0**	95.1	95.8
	Yahoo! Answers: **Examples per Label**					
Model	1	2	4	8	16	32
LSAP						
FT	**49.2**	**58.8**	**60.7**	**63.3**	64.7	**66.4**
LDGT	39.4	54.1	58.4	62.2	**65.3**	**66.4**
	AG News: **Examples per Label**					
Model	1	2	4	8	16	32
LSAP						
FT	**74.8**	**77.2**	**80.7**	**82.3**	**85.4**	**86.5**
LDFT	68.1	72.0	76.0	81.1	84.0	86.0

Table 15: Mean intent classification accuracies for normal fine-tuning (FT; as in §4.1) and label denoising fine-tuning (LDFT) across 5 seeds on Snips, ATIS, Yahoo! Answers, and AGNews at various few-shot split sizes. Smaller splits are subsets of larger splits.

G Hyperparameters

Our experiments are based on the huggingface implementation (Wolf et al., 2020) of T5.

For secondary pre-training, we use initial learning rate 5×10^{-4} and batch size 128. We tune over the number of training epochs $\in [1, 8]$, finding 3 epochs to generally be best.

During fine-tuning, we use init. LR 5×10^{-4} and batch size 1.[13] We tune over the number of fine-tuning epochs $\in [1, 16]$ for the largest few-shot split, typically finding 2 epochs to be best. Once we have the best setting for the largest split, we double the number of tuning epochs for each halving of the split size such that the number of tuning steps is similar for all split sizes. All other hyperparameters are huggingface defaults.

[13] We use a small batch size due to the small size of the 1-shot splits. We do not observe significant performance differences when using batch size 2 or 4.

	TOPv2 (reminder): **Ex. per Label**			
Model	1	2	4	8
LSAP				
FT	**69.0**	**71.3**	**80.6**	87.7
LDFT	65.8	69.3	74.6	86.2
	TOPv2 (weather): **Ex. per Label**			
Model	1	2	4	8
LSAP				
FT	**72.7**	77.4	**81.4**	83.4
LDFT	70.6	**78.4**	80.8	**83.5**

Table 16: Mean intent classification accuracies for normal fine-tuning (FT; as in §4.1) and label denoising fine-tuning (LDFT) across 5 seeds on TOPv2 (*reminder* and *weather* domains) at various few-shot split sizes. Smaller splits are subsets of larger splits.