Label Representations in Modeling Classification as Text Generation

Xinyi Chen[*]
New York University
xc1121@nyu.edu

Jingxian Xu[*]
New York University
jx880@nyu.edu

Alex Wang
New York University
alexwang@nyu.edu

Abstract

Several recent state-of-the-art transfer learning methods model classification tasks as text generation, where labels are represented as strings for the model to generate. We investigate the effect that the choice of strings used to represent labels has on how effectively the model learns the task. For four standard text classification tasks, we design a diverse set of possible string representations for labels, ranging from canonical label definitions to random strings. We experiment with T5 (Raffel et al., 2019) on these tasks, varying the label representations as well as the amount of training data. We find that, in the low data setting, label representation impacts task performance on some tasks, with task-related labels being most effective, but fails to have an impact on others. In the full data setting, our results are largely negative: Different label representations do not affect overall task performance.

1 Introduction

State-of-the-art transfer learning methods model classification tasks as text generation, such as GPT2 (Radford et al., 2019) and T5 (Raffel et al., 2019), and have led to significant improvements across a variety of NLP tasks. In this setting, labels are represented as strings for the model to generate, and the pretrained language model is finetuned to maximize the probability of generating the chosen label representation. For example, for CoLA (Warstadt et al., 2019) task, a classifier will be trained to output 0 (representing unacceptable) or 1 (representing acceptable), while T5 models it as a generation task, it is trained to generate the string "unacceptable" or "acceptable". The advantage of this approach is that the language model can be applied to the classification task as-is, without the need for any additional task-specific parameters or training.

However, in this setting, the impact of the particular strings used to represent labels remains unclear on the end task performance. One of the few studies on this question find that the linguistic properties (relatedness, polarity scale, etc.) of the labels do affect task performance (Nogueira et al., 2020), though their results are limited to document retrieval. We therefore further investigate the impact of string representation when modeling text classification as text generation.

We experiment with T5-base and four diverse, standard text classification tasks. For each task, we design a wide range of label representations, including canonical task labels; task-unrelated antonyms; and completely random strings. As previous works by Nogueira et al. (2020) have noted that the impact of label representation is particularly noticeable in lower data settings, we also vary the amount of training data for each task.

Our experiments reveal that, in the full data setting, the choice of label representation largely does not affect overall performance, with only one of the four datasets seeing any impact. In the low data setting, label representations sometimes have an impact on the overall performance, with task-related labels being the most effective in these cases.

2 Related Work

Target Word Probing Experiments Nogueira et al. (2020) probe the effects of label representation on document retrieval and ranking. They set the baseline mapping as {Positive → `true`, Negative → `false`}, and also try the reverse mapping, antonyms, related words, unrelated words and subwords. In the low data setting, they find that the baseline mapping yields accuracy significantly higher than other types of mappings do. In the high

[*] These two authors contributed equally.

data setting, related words mapping is the most effective, though the differences between mappings are not as large as in the low-data regime. We extend these experiments with more diverse tasks and label representations.

Cloze Reformulation Schick and Schütze (2020) introduce Pattern Exploiting Training, where the input is transformed into a Cloze-style sentence. For example, the task of identifying whether two sentences a and b contradict or agree with each other is reformulated into "a? [blank], b.", and a pretrained language model is used as-is to generate `Yes` or `No` to fill in the blank. They claim that this procedure significantly improves the performance on several tasks in the zero-shot setting. Similarly, Petroni et al. (2019) probe the knowledge presented in state-of-the-art language models without fine-tuning using similar sentences with blanks. They find that these language models contain rich factual knowledge from pretraining, and are effective at recalling knowledge when answering fact-related questions.

Prompt Design Jiang et al. (2020) propose several methods to automatically generate efficient prompts for extracting knowledge from pretrained models, rather than manually designing them. They find that different templates, e.g. "x who converted to y", compared to "x is affiliated with y religion", can improve accuracy by as much as 60%. This line of work is orthogonal to ours: We focus on optimal label representation whereas they focus on the best way to format inputs.

3 Label Representations

We consider four standard text classification datasets representing different tasks and textual genres: (i) sentence acceptability judgments with CoLA (Warstadt et al., 2019); (ii) sentiment analysis with SST-2 (Socher et al., 2013); (iii) paraphrase detection with PAWS (Zhang et al., 2019); and (iv) commonsense reasoning with COPA (Roemmele et al., 2011). For each of these datasets, we test a wide variety of label representations.

3.1 Random Labels

As a simple baseline, we test whether the labels need to be semantic at all. We focus on random labels with short lengths, e.g. {`unacceptable` → `i`, `acceptable` → `c`}.

3.2 Task-Unrelated Labels

Next, we choose sets of words for labels based on their relationship to the task and to each other. We generate them by the following different rules:

- **Antonyms:** We choose words that are antonyms but are semantically unrelated to our tasks, e.g. {`unacceptable` → `cold`, `acceptable` →`hot`}. This setting tests whether it is important that labels be task related or if it is sufficient that they have opposing meanings.
- **Synonyms:** To contrast the antonyms, we use words that are synonyms but are semantically unrelated to our tasks, e.g. {`unacceptable` → `cold`, `acceptable` → `chilly`}.
- **Irrelevant words:** We choose words that are not related to the task nor each other, e.g. {`unacceptable` → `ice`, `acceptable` → `happy`}.
- **Relevant words:** We pick words that are relevant to each other but are not antonyms or synonyms, e.g. {`unacceptable` → `apple`, `acceptable`: → `orange`}.

3.3 Task-Related Labels

We further study how much performance varies between semantically similar task-related label representations. We choose sets of words as labels that have the same polarity scale or meanings as the original targets, e.g. {`unacceptable` → `no`, `acceptable` → `yes`}; As an additional baseline, we use labels that have the opposite polarity or meaning as the original labels, e.g. {`unacceptable` → `yes`, `acceptable` → `no`}.

4 Experiments and Results

4.1 Model and Optimization

Inspired by Nogueira et al. (2020), we experiment with T5 (Raffel et al., 2019), specifically the T5-base model. Among the four datasets, PAWS is the only dataset that has not been pretrained on by T5. So, for CoLA, SST-2, and COPA, we format each example the same as by T5. For PAWS, because it is the same task as MRPC (Dolan and Brockett, 2005), we format the examples identically to how Raffel et al. (2019) process MRPC. See Table 1 for examples.

(CoLA) **input**: `cola sentence: John made Bill master of himself.` **target**: `acceptable`
(SST-2) **input**: `sst2 sentence: it confirms fincher 's status as a film maker who artfully bends technical know-how to the service of psychological insight.` **target**: `positive`
(COPA) **input**: `copa choice1: Many citizens relocated to the capitol. choice2: Many citizens took refuge in other territories. premise: Political violence broke out in the nation. question: effect` **target**: `choice2`
(PAWS) **input**: `paws sentence1: In the early years , KETS was associated with National Educational Television , the forerunner of the current PBS. sentence2: In the early years , KETS was connected with National Educational Television , the forerunner of the current PBS.` **target**: `equivalent`

Table 1: Input formatting for all tasks, with the original canonical label representations from Raffel et al. (2019).

For COPA, CoLA, and SST-2, we finetune with optimize Adam (Kingma and Ba, 2014) for 2000 steps with learning rate 10^{-4}. For PAWS, we finetune for 3000 steps and similarly evaluate the accuracy of the development set every 200 steps. For all datasets, we evaluate the accuracy of the development set every 200 steps and report the best accuracy. For each dataset and setting, we evaluate three runs with different random seeds.

4.2 Low-Data Settings

As Nogueira et al. (2020) found significant differences in performance with distinct data sizes, we also run all four tasks in lower-data regimes. We choose the datasets mainly following these rules: For all tasks, we use the full training sets. Then, we downsample the training set to sizes for which we observe larger gaps between different labels than at the full dataset size. For each run with the low-data setting, we randomly generate a sample before training it, so the sample set normally varies for each run. To test an extremely low-data setting, for COPA and SST-2, we also choose a dataset with an absolute size of 10.

4.3 Results

We present results for SST-2 and COPA in Table 2 and for PAWS and CoLA in Table 3. In the full data setting, we obtain performance near that of BERT (Devlin et al., 2018) for all tasks. It is notable that even in the extremely low data settings, we obtain nontrivial performance for all tasks except COPA, which is likely due to the amount of pretraining in even the T5-base model.

For SST-2, by training on the full dataset, we get similar results for all choices of label representation. Notably, even the random strings perform as well as the task-specific labels. However, given the large size of the full dataset, it is unsurprising that the model can learn the class definitions from random strings.

For the 100 and 10 example settings, we find that the original labels achieve the best accuracy, 2% less than the full data setting. However, a potential confounder is the fact that T5 was pretrained on SST-2 with these labels, which likely explains the high accuracy with only 10 examples. We find the mutually unrelated and task-unrelated labels perform as well as the original labels in the 100 example setting (`ice`/`happy`), suggesting that even with this few examples, the choice of labels or even the relationship between them is not crucial. Similarly, in the 10 example setting, these mutually unrelated labels (`ice`/`happy`) perform as well as task-specific labels and task-unrelated antonyms, providing further evidence that the choice of labels is not crucial to learning the task, even in extremely low data settings. On the other hand, reversing the original labels or reversing task-related labels consistently performs the worst, even worse than random labels. This suggests that the label representations do not matter, as long as we do not pick labels that flip the class definition.

For COPA, in the full data setting, we observe notable performance differences between various labels. We get the best performance using the original labels and other task-related labels, while task-unrelated labels generally perform much worse. For the random labels, we observe high variability,

		SST-2			COPA		
label type / training set size		67349 (100%)	100 (0.167%)	10 (0.015%)	400 (100%)	80 (20%)	10 (2.5%)
original		94.48 ± 0.37	93.83 ± 0.29	93.92 ± 0.00	68.00 ± 2.00	65.00 ± 1.00	58.33 ± 0.58
reversed		94.07 ± 0.54	89.46 ± 1.00	31.54 ± 15.10	66.00 ± 3.61	63.67 ± 8.74	52.67 ± 5.13
random	i/c	94.38 ± 0.72	91.96 ± 0.90	72.36 ± 3.74	59.00 ± 5.00	54.67 ± 0.58	52.00 ± 4.00
	n/p	93.96 ± 0.07	91.67 ± 0.76	73.43 ± 3.46	68.33 ± 3.51	62.67 ± 2.52	54.67 ± 4.73
related	matched	94.50 ± 0.23	92.00 ± 0.87	75.96 ± 0.29	69.67 ± 2.31	68.33 ± 2.08	51.33 ± 9.24
	reversed	94.46 ± 0.24	90.50 ± 2.18	54.01 ± 0.53	69.33 ± 3.79	57.00 ± 6.00	43.67 ± 2.31
unrelated	cold/hot	94.75 ± 0.15	92.17 ± 0.76	84.29 ± 3.00	62.67 ± 2.52	55.00 ± 1.00	46.67 ± 2.08
	cold/chilly	94.19 ± 0.18	91.86 ± 0.92	73.05 ± 7.47	67.00 ± 2.65	61.33 ± 1.53	52.67 ± 1.53
	apple/orange	94.00 ± 0.18	91.17 ± 1.04	65.14 ± 2.56	55.67 ± 2.08	53.33 ± 2.08	52.00 ± 3.46
	ice/happy	94.69 ± 0.18	93.67 ± 0.58	84.29 ± 3.51	56.33 ± 4.04	54.33 ± 4.73	49.00 ± 3.00

Table 2: Accuracy (mean ± std) for SST-2 and COPA for a variety of dataset sizes and label representations. The original labels for SST-2 and COPA are 'negative/positive' and 'choice1/choice2', respectively. Matched labels for SST-2 and COPA are 'no/yes' and 'first/second', respectively.

		PAWS			CoLA		
label type / training set size		49401 (100%)	4000 (8.10%)	400 (0.81%)	8551 (100%)	4000 (46.8%)	400(4.68%)
original		94.09 ± 0.14	91.48 ± 0.24	83.50 ± 0.54	58.64 ± 0.15	57.15 ± 0.30	48.67 ± 0.00
reversed		94.68 ± 0.06	91.48 ± 0.24	85.04 ± 0.75	56.58 ± 1.95	55.78 ± 1.22	33.48 ± 6.74
random	i/c	93.00 ± 0.36	91.02 ± 0.26	82.50 ± 4.05	58.53 ± 1.28	56.70 ± 0.53	49.28 ± 3.16
	n/p	93.58 ± 0.44	91.49 ± 0.55	83.50 ± 1.82	57.14 ± 0.95	57.81 ± 1.75	46.85 ± 1.25
related	matched	93.63 ± 0.23	91.08 ± 0.45	84.83 ± 0.51	57.47 ± 2.43	55.59 ± 1.17	41.30 ± 4.23
	reversed	93.42 ± 0.51	92.81 ± 1.22	84.38 ± 1.19	57.22 ± 0.88	55.82 ± 0.39	28.33 ± 1.70
unrelated	cold/hot	93.12 ± 0.47	91.73 ± 0.53	83.50 ± 2.10	57.53 ± 0.52	56.18 ± 1.14	39.21 ± 2.39
	cold/chilly	93.55 ± 0.54	91.30 ± 0.19	84.28 ± 1.56	58.57 ± 3.01	57.47 ± 0.41	45.27 ± 1.58
	apple/orange	93.85 ± 0.21	91.29 ± 0.27	82.50 ± 0.99	58.77 ± 1.48	56.48 ± 1.27	33.38 ± 1.32
	ice/happy	93.16 ± 0.13	91.67 ± 0.38	83.00 ± 2.38	56.97 ± 1.80	56.34 ± 1.26	47.64 ± 2.67

Table 3: Accuracy (mean ± std) for PAWS and Matthews' correlation coefficient for CoLA for a variety of dataset sizes and label representations. The original labels for PAWS and CoLA are 'not_equivalent/equivalent' and 'unacceptable/acceptable', respectively. Matched labels for PAWS and CoLA are 'different/same' and 'no/yes', respectively.

with one pair of labels (n/p) worryingly performing as well as the original task labels. Also concerning is the fact that reversing the original or other task-related labels performs nearly as well as not reversing them. We speculate that the small dataset size contributes to these odd performance trends, as has been previously noted for COPA (Sap et al., 2019).

In the extremely low data settings, differences between labels become more significant and noticeable, with several label representations failing to learn the task and obtaining accuracy near chance (50%). The original and reversed pairs, as well as the task-related pairs, show similar accuracies, and these labels generally perform the best among all labels. Similar to SST-2, reversing the original labels performs worse and shows more instability. In the smallest setting, matched task-related labels also perform better than reversed ones and all other labels except for the original label pair.

For CoLA and PAWS, in full data and 4000 sample data regimes, the performances we get using different labels vary, but by no more than 2%. When the data size is extremely low, we observe notable gaps among labels for CoLA, but not for PAWS. Similar to SST-2, for CoLA, we find that reversed labels perform much worse than original ones.

5 Conclusion

In this work, we investigate the impact of label representation in modeling classification as a seq2seq task. For four standard text classification datasets and task types, we design a wide range of label representations, ranging from canonical task-related labels to task-unrelated antonyms to random words and strings. We experiment with the T5-base model on these datasets and label representations in a range of regimes with various data sizes.

Overall, we find that the choice of label representation largely does not affect task performance, though it varies by task and dataset size. In the high data settings, there is generally no differences between choices of label representations, and even random strings can function well as label repre-

sentations. In low data settings, the influence of label representations varies significantly between tasks. For PAWS, we observe no variation, but for COPA and CoLA, we note that task-related labels generally perform best.

Our experiments represent preliminary negative evidence that label representations have limited impact on task performance, but there are a number of dimensions that our work does not investigate that might affect a model's sensitivity to label representation.

First, we take as given that canonical task labels as sufficient for all tasks. However, for some tasks, the canonical label representation might be suboptimal or not sufficiently convey the semantics of the task, e.g. `choice1`/`choice2` for COPA. A natural way to mitigate this issue would be to use a much larger set of possible label representations, or even automatically discover label representations. However, we expect this issue will be especially problematic with more complex tasks that are difficult to represent with single-word labels or to succinctly represent in text at all.

Second, we use a single task input format, but the task formatting may be suboptimal and affect the ability of the model to learn from the label semantics. Schick and Schütze (2020) use multiple example templates per task and find that performance between templates can vary substantially. While an obvious direction would be to simply use multiple templates, automatically discovering effective templates also seems like a promising direction.

Acknowledgements

We thank Samuel R. Bowman, Nikita Nangia, and Marina Zavalina for their guidance and support; Yang Liu for his mentoring and advice.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document Ranking with a Pretrained Sequence-to-Sequence Model. *arXiv e-print 2003.06713*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473.

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. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv e-print 1910.10683*.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *EMNLP*.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv e-print 2001.07676*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

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

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. Paws: Paraphrase adversaries from word scrambling.