# Task-Aware Representation of Sentences for Generic Text Classification

**Kishaloy Halder**[†] **Alan Akbik**[‡] **Josip Krapac**[†] **Roland Vollgraf**[†]
[†] Zalando SE
[‡] Humboldt-Universität zu Berlin

kishaloy.halder@zalando.de alan.akbik@hu-berlin.de
josip.krapac@zalando.de roland.vollgraf@zalando.de

## Abstract

State-of-the-art approaches for text classification leverage a transformer architecture with a linear layer on top that outputs a class distribution for a given prediction problem. While effective, this approach suffers from conceptual limitations that affect its utility in few-shot or zero-shot transfer learning scenarios. First, the number of classes to predict needs to be pre-defined. In a transfer learning setting, in which new classes are added to an already trained classifier, all information contained in a linear layer is therefore discarded, and a new layer is trained from scratch. Second, this approach only learns the semantics of classes *implicitly* from training examples, as opposed to leveraging the *explicit* semantic information provided by the natural language names of the classes. For instance, a classifier trained to predict the topics of news articles might have classes like "business" or "sports" that themselves carry semantic information. Extending a classifier to predict a new class named "politics" with only a handful of training examples would benefit from both leveraging the semantic information in the name of a new class and using the information contained in the already trained linear layer. This paper presents a novel formulation of text classification that addresses these limitations. It imbues the notion of the task at hand into the transformer model itself by factorizing arbitrary classification problems into a generic binary classification problem. We present experiments in few-shot and zero-shot transfer learning that show that our approach significantly outperforms previous approaches on small training data and can even learn to predict new classes with no training examples at all. The implementation of our model is publicly available at: https://github.com/flairNLP/flair.

## 1 Introduction

Text classification is the task of predicting one or multiple class labels for a given text. It is used in a large number of applications such as spam filtering (Jindal and Liu, 2007), sentiment analysis (Rosenthal et al., 2017), intent detection (Hollerit et al., 2013) or news topic classification (Zhang et al., 2015). The current state-of-the-art approach to text classification leverages a BERT-style transformer architecture (Devlin et al., 2019; Yang et al., 2019; Lan et al., 2020) with a linear classifier layer on top. The transformer is pre-trained on language modelling task, whereas the classifier is randomly initialized. The entire model is then fine-tuned using training examples for all classes, so that the classifier outputs a distribution over all class labels in the prediction problem. This approach is shown to work well, especially if for each class a reasonable amount of training examples is available.

**Few-shot transfer learning.** Real world text classification scenarios are often characterized by a lack of annotated corpora and rapidly changing information needs (Chiticariu et al., 2013), motivating research into methods that allow us to train text classifiers for new classes with only a handful of training examples (Bansal et al., 2019; Yogatama et al., 2019). In such cases, a standard approach is to transfer knowledge from an existing model for classification task $X$ to initialize the weights for a model for the new classification task $Y$. Here, there are two options: If task $Y$ differs from $X$ significantly, then we





*Proceedings of the 28th International Conference on Computational Linguistics*, pages 3202–3213
Barcelona, Spain (Online), December 8-13, 2020

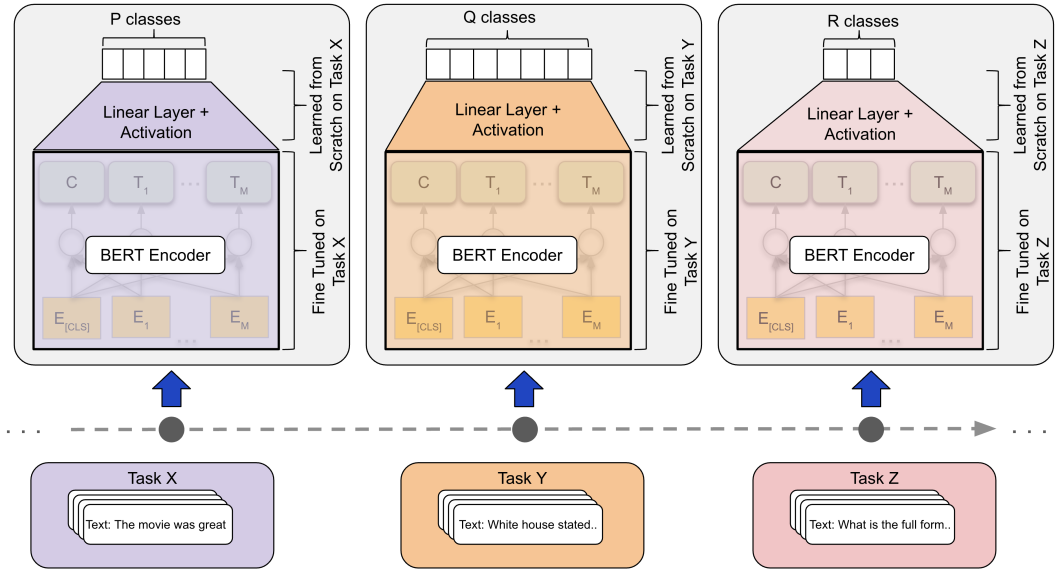


Figure 1: Traditional text classifier model training on different tasks *X*, *Y* and *Z*. While the BERT-encoder can be transferred between tasks, the final linear layer and activation are task-specific and therefore must be learned separately from scratch for each task. This limits the transfer learning capability.

might discard the entire model that was learned for task *X*, and learn a model afresh. In case task *Y* is somewhat similar to *X*, then the fine-tuned BERT encoder can potentially be transferred to act as the starting point for task *Y*. This is illustrated in Figure 1.

However, this approach to transfer learning disregards or dismisses two sources of information that may be especially useful in few-shot or zero-shot scenarios:

**Information in the pre-trained decoder** First, the traditional transfer learning approach will discard the final linear layer that acts as decoder, since the prediction targets might differ *i.e.,* $P \neq Q$ or there is no one-to-one correspondence (see Figure 1). This effectively results in a loss of all information contained in the decoder and requires us to train a new decoder from scratch given very limited training data in a *few-shot* scenario. Worse, this approach cannot be used in a *zero-shot* scenario at all since here there is no training data to train the decoder.

**Information provided by class labels** Second, the traditional approach only learns the semantics of classes *implicitly* from their training examples. This disregards the *explicit* semantic information provided by the natural language class labels. For instance, a classifier trained to predict the topics of news articles might have class labels like "business" or "sports" that themselves carry semantic information (Meng et al., 2018; Puri and Catanzaro, 2019). If such a classifier were to be extended to predict a new class named "politics" with only a handful of training examples, it may be sensible to leverage the semantics provided by this class label as well. This would extend its theoretical applicability to zero-shot learning since the name of the new class could suffice as input to learn new classifiers, even without training data.

With this paper, we present a straightforward but remarkably effective approach to preserve the two above-mentioned sources of information in transfer learning for text classification. The main idea is to imbue the notion of the task itself into the transformer model, by factorizing arbitrary classification problems into a generic binary classification problem. In other words, we replace the task-specific decoder with a generic binary "True/False" decoder. The input to the transformer then consists not only of the text to be classified, but also of the class label (e.g. a semantically meaningful form of textual labels) prepended to the text. We illustrate this in Figure 2.

**Task-Aware Representations.** Our proposed approach therefore reformulates the classification problem as a "query" in which a sentence and a potential class label is given to the transformer which makes a pre-

diction whether or not this label holds. The cross-attention mechanism in BERT then learns to combine the representation of the text and its label. Accordingly, we refer to this approach as TARS (**T**ask-**A**ware **R**epresentation of **S**entences). This addresses the two issues mentioned above as the same decoder can now be used across arbitrary tasks (allowing transfer of the full model) and that the information provided by the class label itself is interpreted by the transformer model. A conceptual advantage of this approach is that it can return predictions even for classes for which no training data exists: it simply needs to prepend the textual label of the new class to text and evaluate the result of the "True/False" decoder. Our contributions are therefore as follows:

1. We present TARS, a novel formulation of text classification to address crucial shortcomings of traditional transfer learning approaches, and show how TARS can be trained seamlessly across tasks. We also illustrate how TARS can learn in a zero-shot scenario.

2. We conduct an extensive evaluation of TARS' zero-shot and few-shot transfer learning abilities using text classification datasets from different tasks (sentiment analysis, topic detection, question type detection) and different domains (newswire, restaurant reviews, product reviews) to compare against baselines and investigate the impact of semantic distance.

3. We release all code to the research community for reproduction of our experiments integrated with FLAIR[1] framework.

We find that TARS significantly outperforms traditional transfer learning in regimes with little to no training data. We also observe surprisingly powerful zero-shot learning abilities, indicating that TARS indeed learns to interpret the semantics of the label name and is thus able to correctly predict labels for classes without any training data at all. Based on these results, we conclude TARS to be a conceptually simple and effective approach for few-shot and zero-shot transfer learning in text classification.

## 2 Method

We formulate text classification as a universal binary classification problem and use cross-attention to capture the modified objective. We then illustrate how we train, predict and transfer using TARS.

### 2.1 Universal Binary Text Classification Formulation

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

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

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

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

$$f :< task\, label, text > \rightarrow \{0,1\} \quad \textit{i.e.,} \quad f(\text{label}(y_i), t) = P(\text{ True} \,|\, y_i, t) \, \forall \, i \in \{1...M\} \tag{2}$$

In other words, we provide a tuple consisting of both the text input as well the class label name to the function and ask if there is a *match*. For example, input to a binary sentiment classifier (trained to predict whether a text has *positive* or *negative* sentiment) would consist both of the text to be classified as well as the possible label:

```
<"positive sentiment", "I enjoyed the movie a lot">
```

[1] Available in FLAIR (Akbik et al., 2019) version 0.7 onward.

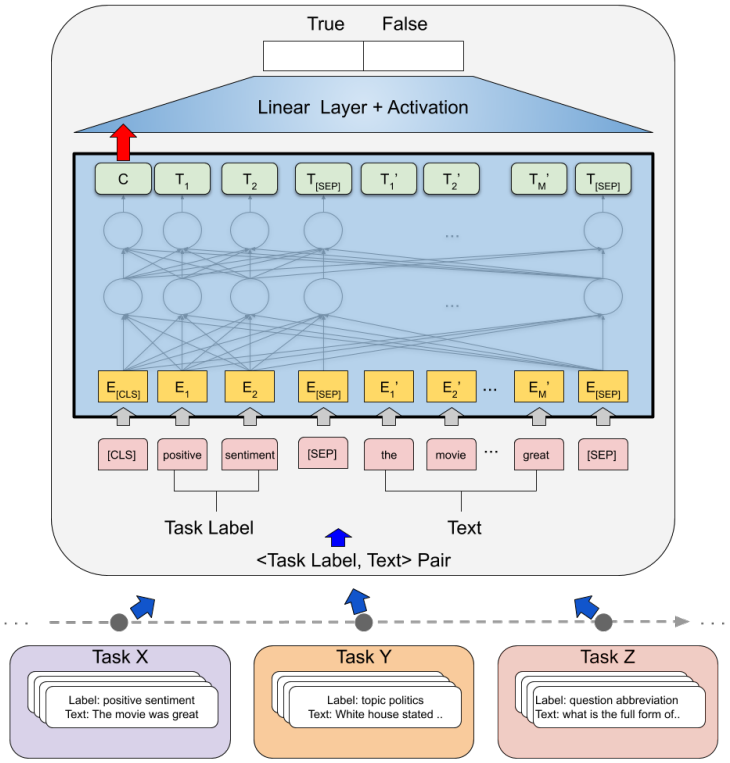


Figure 2: Architecture of our proposed TARS model. Input is a tuple of both the text and a possible label and the output across all tasks is a True/False prediction. Since the architecture remains identical, the same model may be used across any task: Training this model on a new task is equivalent to continuing to train an existing model on new data.

With the output being either True or False depending on whether this label matches the text. Similarly for topic classification in which topics labels such as "politics" or "sports" should be predicted, an example for an input tuple is the following:

```
<"topic politics", "The White House announced that [..]">
```

As can be seen from the above illustrations, any classification task can be factorized into our definition of the problem. This resembles decomposition of multi-class classifications into multiple binary classification which has been studied in early machine learning literature (Aly, 2005; Allwein et al., 2000), but differs in that we define a function approximation method that can unify many text classification problems into one, and yield a *single* model to perform predictions after due training.

### 2.2 Cross-Attention between Text and Label

Since we replace the linear layer with a binary True/False choice, we effectively impoverish the decoder while providing additional input to the BERT-encoder in the form of the class label. This means that the encoder itself must learn to understand the connection between a class label and a given text. Here, we make use of the cross-attention mechanism that transformer architectures supply. This is trivially accomplished by prepending the class label to the text using the special separator token `[SEP]` commonly used in BERT. So, our input sequence consists in this order of the `[CLS]`-token, the class label, the `[SEP]`-token and the text to classify. This input sequence is then passed through all self-attention layers in BERT. In keeping with prior work we use the representation of the `[CLS]`-token in the final layer as the task label dependent representation of the input text. This is illustrated in Figure 2.

On top of this encoder stack, we use a linear layer to project the $H$-dimensional tensor produced by the encoder into 2 real-valued logits. A *softmax* function is used to form a probability distribution over 2 classes *i.e.,* True, and False.

### 2.3 Training and Prediction

In our formulation, we populate $M$ <task label, text> pairs for each sample text for a text classification task with $M$ classes. For example, if the ground truth label for $j^{\text{th}}$ input ($t_j$) is $i^{\text{th}}$ class ($y_i$), then in our transformed formulation we have ($<$ label$(y_i), t_j >$, True) as one training sample, and ($<$ label$(y_k), t_j >$, False) whenever $k \neq i$, leading to another $M - 1$ samples for our TARS model. To illustrate, the above used example tuple for sentiment analysis needs to be rephrased to *two* input/output pairs during training, one for each sentiment label:

<"positive sentiment", "I enjoyed the movie a lot"> → TRUE
<"negative sentiment", "I enjoyed the movie a lot"> → FALSE

This effectively increases the amount of the training data and thus the computational costs by a factor of $M$ and is the main conceptual drawback of TARS. We use a similar approach during prediction in that tuples for all possible labels $M$ are populated and passed through the model to obtain $M$ True/False predictions. For multi-class problems we use the class with maximum confidence (for True) as the final prediction:

$$\hat{y} = \underset{k \in \{1 \ldots M\}}{\arg\max}\ f(\text{label}(y_k),\ x) \tag{3}$$

To train the model, we follow standard practice and use *cross-entropy* loss, and optimize all parameters using gradient descent.

### 2.4 Model Transfer

The core advantage of TARS is that the entire model (encoder and decoder) can be shared across tasks, as the encoder now performs the matching between label and text. This means that using transfer learning to train a new tasks becomes equivalent to continuing to train the same model with different training data. As we show in the experiments section, this holds advantages in few-shot learning scenarios. If there is enough similarity between tasks (*e.g.,* the nature of the classification task, and/or word distributions), this formulation even enables a zero-shot scenario provided the transformer is able to correctly interpret the semantics of the new class label.

Beyond transfer learning, our formulation also trivially enables multi-task learning across corpora with different annotations as we do not require separate prediction heads for each task. Rather, we can train the same model using <task label, text> tuples from different tasks (see Figure 2) and during prediction only request predictions for the labels we require.

### 2.5 Computational complexity

While traditional text classification (*cf.* Figure 1) requires one forward pass per task and input to obtain predictions for all $M$ classes, TARS (*cf.* Figure 2) requires $M$ forward passes, one for each class-input pair. On the other hand, the model parameters for different tasks are shared, so only one model for all tasks is kept in memory, while traditional models require a separate model for each task. Therefore TARS is more suited for training many tasks, with small number of labels and small amount of data per label. In Section 5 we discuss future strategies to address the computational complexity.

## 3 Experiments

We conduct an experimental evaluation of TARS to address the following questions: (1) How well is TARS able to transfer to new classification tasks with little training data? (2) How does semantic distance between source and target task affect the transfer learning abilities of TARS? (3) And what are the zero-shot capabilities of TARS?

**Datasets and labels.** To this end, we experiment with 5 widely-used text classification datasets that span different textual domains and classification tasks: Two datasets for the task of topic detection, namely AGNEWS (Zhang et al., 2015), a corpus of news articles classified into 4 topics, and DBPEDIA (Zhang et al., 2015), a corpus of 14 entity topics. One dataset in two variants for the task of classifying question

types (Li and Roth, 2002), namely TREC-6 with 6 coarse-grained and TREC-50 with 50 fine-grained question types. And two corpora for 5-class sentiment analysis, namely AMAZON-FULL (Zhang et al., 2015) for product reviews and YELP-FULL (Zhang et al., 2015) for restaurant reviews. An overview of all 6 datasets is given in Table 1.

| Dataset | Type | #classes | #train | #test | avg #chars | avg #words |
|---|---|---|---|---|---|---|
| TREC-6 (Li and Roth, 2002) | Question | 6 | 5.5k | 500 | 60 | 11 |
| TREC-50 (Li and Roth, 2002) | Question | 50 | 5.5k | 500 | 60 | 11 |
| YELP-FULL (Zhang et al., 2015) | Sentiment | 5 | 650k | 50k | 735 | 136 |
| AMAZON-FULL (Zhang et al., 2015) | Sentiment | 5 | 1.19M | 630k | 450 | 80 |
| AGNEWS (Zhang et al., 2015) | Topic | 4 | 120k | 7.6k | 241 | 37 |
| DBPEDIA (Zhang et al., 2015) | Topic | 14 | 560k | 70k | 304 | 49 |

Table 1: Dataset statistics.

In case of topic classification on AGNEWS, and DBPEDIA we have short class labels available. In some cases, we manually curated terse labels so that they form individual words e.g., "Sci/Tech" was renamed to "Science Technology", "EducationalInstitution" to "Educational Institution" and so on. For the sentiment analysis datasets, a numeric rating (*i.e.,* $1-5$) is available along with each sample. We use some textual descriptions for them instead of relying on the numeric rating[2].

**Transfer learning setup.** Our setup distinguishes a *source task* and a *target task*. The model for the source task is trained using the full dataset for the respective task. To evaluate transfer learning capabilities in few-shot and zero-shot scenarios, we then fine-tune the source model on the target task using only very limited numbers of training examples. We report accuracy for all the baseline models for different transfer scenarios. To evaluate how quickly the models adapt to the new target task, we increase the number of training examples per class ($k$) seen by the model. We start with *zero shot* scenario, where the model does not see *any* training example from the target task (*i.e.,* $k = 0$). Then we expose the models to increasing number of randomly chosen samples per class from the target task ($k = 1, 2, 4, ...$), and observe how fast the competing models are able to leverage new labeled data. In all cases, we evaluate the baseline models on the *entire* test data available.

**Baselines.** We compare TARS against two baselines:

- **BERT$_{\text{BASE}}$:** This is the standard non-transfer learning variant in which we fine-tune a pre-trained BERT-model ('`bert-base-uncased`'[3]) with a linear classifier on top directly on the target task.
- **BERT$_{\text{BASE}}$ (ft):** In this variant, we first fine-tune BERT on the source task. We then transfer the encoder weights to a new model and initialize a new linear layer, and fine-tune this model again on the target task. This covers the traditional transfer learning mechanism prevalent in the literature.

Both baselines assume multi-class setting and use multinomial logistic regression (*softmax* function at classifier output). We use the FLAIR library to implement the baselines (Akbik et al., 2018), setting a batch size of 16, a learning rate of $0.02$, and a maximum number of 20 epochs, after which we follow standard practice to select the best model based on development holdout data. We use the pre-trained tokenizer available with the '`bert-base-uncased`' model. In rare cases where the input sequence is longer than $512$ subtokens, they are truncated. Since transformer models are sensitive to the choice of random seed, we repeat each experiment $5$ times with different random seeds and report the average accuracy along with the standard deviation.

### 3.1 Results

Table 2 presents the results of *in-domain* transfer learning for source and target data pairs that are of the same broad category of classification task. That is, we evaluate transfer between the two sentiment analy-

[2] `https://kishaloyhalder.github.io/pdfs/tars_appendix.pdf`
[3] `https://github.com/huggingface/transformers`

| Domain: Sentiment Analysis | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| YELP-FULL → AMAZON-FULL | | | | | AMAZON-FULL → YELP-FULL | | | | |
| $M$ | $k$ | $\text{BERT}_{\text{BASE}}$ | $\text{BERT}_{\text{BASE}}$ (ft) | TARS | $M$ | $k$ | $\text{BERT}_{\text{BASE}}$ | $\text{BERT}_{\text{BASE}}$ (ft) | TARS |
| | 0 | – | – | **51.8** | | 0 | – | – | **50.6** |
| | 1 | 21.8±1.7 | 27.5±6.5 | **51.0**±0.3 | | 1 | 22.5±3.2 | 28.0±5.3 | **53.0**±0.3 |
| | 2 | 24.6±1.1 | 36.4±7.0 | **52.7**±0.2 | | 2 | 22.6±1.7 | 33.7±4.1 | **52.2**±0.7 |
| 5 | 4 | 25.8±1.7 | 43.2±3.0 | **52.3**±0.5 | 5 | 4 | 26.5±2.3 | 44.1±1.4 | **52.0**±2.1 |
| | 8 | 25.4±1.8 | 45.0±1.1 | **49.9**±1.7 | | 8 | 31.9±2.0 | 46.5±2.0 | **53.3**±1.1 |
| | 10 | 29.0±1.5 | 45.2±1.0 | **51.6**±0.4 | | 10 | 32.8±2.1 | 47.2±3.0 | **52.5**±0.3 |
| | 100 | 50.7±0.9 | 53.2±0.4 | **53.4**±0.4 | | 100 | 53.9±1.8 | 55.8±0.5 | **56.4**±0.7 |
| Domain: Topic Classification | | | | | | | | | |
| DBPEDIA → AGNEWS | | | | | AGNEWS → DBPEDIA | | | | |
| $M$ | $k$ | $\text{BERT}_{\text{BASE}}$ | $\text{BERT}_{\text{BASE}}$ (ft) | TARS | $M$ | $k$ | $\text{BERT}_{\text{BASE}}$ | $\text{BERT}_{\text{BASE}}$ (ft) | TARS |
| | 0 | – | – | **52.4** | | 0 | – | – | **51.2** |
| | 1 | 41.6±6.5 | 66.6±4.6 | **72.1**±3.4 | | 1 | 45.4±2.6 | 45.2±3.7 | **76.6**±2.7 |
| | 2 | 56.0±3.3 | 69.8±2.7 | **74.3**±4.5 | | 2 | 76.4±2.4 | 66.0±4.2 | **81.7**±3.8 |
| 4 | 4 | 70.8±5.6 | 78.5±2.3 | **80.2**±0.9 | 14 | 4 | **91.3±0.5** | 84.4±2.7 | 90.1±1.3 |
| | 8 | 78.3±1.3 | 80.1±2.1 | **81.0**±0.8 | | 8 | **96.5**±0.4 | 93.5±1.4 | 94.8±0.7 |
| | 10 | 80.1±2.9 | 82.0±0.6 | **83.5**±0.2 | | 10 | **97.6**±0.3 | 95.8±0.1 | 96.6±0.2 |
| | 100 | **87.8**±0.4 | 86.9±0.4 | 86.7±0.3 | | 100 | **98.7**±0.0 | 98.4±0.0 | 98.4±0.0 |

| Domain: Question Type Classification | | | | |
|---|---|---|---|---|
| TREC-6 → TREC-50 | | | | |
| $M$ | $k$ | $\text{BERT}_{\text{BASE}}$ | $\text{BERT}_{\text{BASE}}$ (ft) | TARS |
| | 0 | – | – | **53.4** |
| | 1 | 11.4±3.7 | 40.2±4.8 | **57.2**±1.0 |
| | 2 | 29.1±4.7 | 74.5±1.4 | **82.0**±2.6 |
| 50 | 4 | 47.9±5.2 | 78.6±1.3 | **82.7**±2.3 |
| | 8 | 64.4±1.6 | 81.6±1.5 | **86.2**±2.9 |
| | 10 | 67.1±2.9 | 83.2±0.7 | **85.1**±1.0 |
| | 100 | 89.6±0.6 | 91.3±0.2 | **91.4**±0.5 |

| Model | Model Size | AGNEWS | DBPEDIA |
|---|---|---|---|
| GPT-2 (2019) | 117M | 40.2* | 39.6* |
| TARS | 110M | **52.4** | **51.2** |

Table 2: Comparison of baselines on different text classification tasks in terms of accuracy (± standard deviation) on test set in zero/few shot settings. $M$ is the number of classes in target task, $k$ is the number of samples seen per class. TARS consistently outperforms the baselines for very small values of $k$ across domains. *Bottom-right:* TARS also outperforms reported zero shot accuracy scores by a GPT-2 based model of similar size (Puri and Catanzaro, 2019).

sis datasets, the two topic classification datasets and the two question type variants[4]. As described above, we train the target task only using $k$ training examples and compare TARS against our two baselines. We make the following observations:

**Zero-shot classification in TARS far above random baseline.** We firstly find that TARS is successfully able to classify target labels at $k = 0$, i.e. with no target training data at all. We note that in all cases the zero shot accuracy obtained by TARS is considerably higher than random baseline (*e.g.,* 51.2 w/ TARS vs 7.15 w/ random for AGNEWS to DBPEDIA transfer task). In contrast, the baselines are conceptually unable to perform zero shot classification. We also compare TARS against the reported zero-shot accuracy achieved by a GPT-2 based generative model with similar number of parameters on AGNEWS, and DBPEDIA (Puri and Catanzaro, 2019). Although there are some differences in the setup, overall, TARS outperforms it by a wide margin.

**Stronger few-shot results than baselines, but advantage levels off.** We observe that TARS can adapt relatively quickly to the target task, and achieves much higher accuracy scores when all the models get to see very few examples per class. On average, our TARS models achieves a relative gain in accuracy of 24.56%, 9.24%, 6.42% on $k$-shot learning across the transfer tasks with $k = 2, 4, 8$

[4]Note that we evaluate both directions of transfer except for TREC-50 to TREC-6 which would be trivial since TREC-6 is a more coarse-grained variant of TREC-50.

| Cross Domain Transfer | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| DBPEDIA (Topic)→ TREC-6 (Question Type) | | | | | AMAZON-FULL (Sentiment) → AGNEWS (Topic) | | | | |
| $M$ | $k$ | $\text{BERT}_{\text{BASE}}$ | $\text{BERT}_{\text{BASE}}$ (ft) | TARS | $M$ | $k$ | $\text{BERT}_{\text{BASE}}$ | $\text{BERT}_{\text{BASE}}$ (ft) | TARS |
| | 0 | – | – | **43.0** | | 0 | – | – | **28.0** |
| | 1 | 26.4±4.2 | 38.5±3.9 | **45.7**±6.2 | | 1 | **43.8**±4.0 | 29.8±0.7 | 42.9±3.5 |
| | 2 | 36.9±6.0 | 32.8±7.1 | **62.9**±5.7 | | 2 | **59.6**±1.1 | 37.1±4.3 | 49.5±1.0 |
| 6 | 4 | 43.5±3.2 | 45.3±3.0 | **62.7**±2.2 | 6 | 4 | **70.4**±4.6 | 49.0±2.8 | 63.7±6.4 |
| | 8 | 56.4±3.1 | 57.2±1.8 | **61.9**±1.9 | | 8 | **80.5**±0.3 | 57.4±0.8 | 79.2±0.2 |
| | 10 | 58.8±6.6 | 63.7±2.3 | **64.7**±1.0 | | 10 | **81.4**±0.7 | 65.4±6.3 | 79.6±0.7 |
| | 100 | 92.5±0.8 | **93.4**±1.0 | 91.6±0.9 | | 100 | **88.0**±0.1 | 86.9±0.4 | 86.6±0.6 |

Table 3: Comparison of baselines on cross-domain transfer task.

respectively[5]. The relative gains achieved by TARS are higher for the sentiment analysis domain with $49.5\%, 19.45\%, 12.7\%$ compared to that of $10\%, 5.2\%, 5.6\%$ for question type classification, and $6.65\%, 1\%, 0.5\%$ for topic classification with $k = 2, 4, 8$ respectively. We attribute this to differences in linguistic cues between the domains (formal vs. informal). Notice that the baselines are trained with multinomial regression, which takes into account mutual exclusivity of classes at training time. TARS outperforms the baselines without explicitly modeling multi-class assumption at trining time, thus also allowing transfer of knowledge between multi-class and multi-label tasks.

However, we also note that this advantage over $\text{BERT}_{\text{BASE}}$ and $\text{BERT}_{\text{BASE}}$ (ft) levels off as all approaches see more training data. This indicates that our approach is useful mostly in a regime with very little training data. Comparing the baselines, we also note that $\text{BERT}_{\text{BASE}}$ (ft) outperforms $\text{BERT}_{\text{BASE}}$ in most of the cases, showing the effectiveness of the traditional transfer learning approach.

**Effectiveness of transfer learning depends on semantic distance.** Next to in-domain transfer, we evaluate transfer learning between semantically more different datasets. Table 3 shows evaluation results for transfer from topic to question type classification, and from sentiment to topic classification.

We observe that the transfer from DBPEDIA to TREC-6 still shows TARS to significantly outperform the baselines at $k \leq 10$, despite the semantic distance between the two tasks. However, in the transfer from AMAZON-FULL to AGNEWS – two tasks with widely different language and domain – we find that a $\text{BERT}_{\text{BASE}}$ model trained directly on target task data outperforms all transfer learning approaches. Though even here it is interesting to note that TARS still outperforms $\text{BERT}_{\text{BASE}}$ (ft). This speaks to the robustness of TARS in transfer learning even across semantically distant tasks.

### 3.2 Ablation Experiment: Adding a New Class Without Training Data

As outlined in the introduction, a realistic scenario for zero- or few-shot learning is the addition of a new class to an existing classifier. An example is a system that is already able to predict $N$ topic labels to which a new topic is added. A new class addition is the most favorable possible scenario for zero-shot learning since the added label is of the exact same textual domain and semantic class.

To simulate this, we use a subset of 1000 randomly sampled points from the DBPEDIA corpus as source task, but withhold all examples of one class, namely "animal". We then repeat the experiment from above, learning a new model using $k$ examples of the "animal" topic. Since in this setting the class distribution in the seen data is heavily skewed, we report the $f_1$ score for the new class in Table 4. Interestingly, we observe that TARS yields an impressive $f_1$ score of $0.60$ in zero shot, indicating surprisingly high ability to learn the semantics of a new class purely from the class label itself.

**Qualitative inspection.** We qualitatively inspect a sample of text data points correctly and incorrectly classified as "animal", shown in Table 4 (right hand side). For instance, we find that even with no training data at all, the text "*The collared sunbird ( hedydipna collaris ) is a sunbird [...]*" is correctly classified as belonging to class "animal", over 13 other possible topics. This indicates that TARS does indeed learn a correspondence between the natural language task label "animal" and words that occur in the text such

[5]Except for AGNEWS to DBPEDIA transfer where $\text{BERT}_{\text{BASE}}$ adapts faster than its fine tuned variant. We believe that this may stem from $\text{BERT}_{\text{BASE}}$ being pre-trained on Wikipedia (Devlin et al., 2019), the same as DBPEDIA, putting it in advantage.

| Domain: Topic Classification w/ New Class Addition | | | | |
|---|---|---|---|---|
| DBPEDIA-13 → DBPEDIA | | | | |
| $M$ | $k$ | $\text{BERT}_{\text{BASE}}$ | $\text{BERT}_{\text{BASE}}$ (ft) | TARS |
| | 0 | – | – | **0.60** |
| | 1 | 0.05 | 0.40 | **0.72** |
| | 2 | 0.58 | 0.73 | **0.85** |
| 14 | 4 | 0.91 | 0.89 | **0.96** |
| | 8 | 0.93 | 0.91 | **0.95** |
| | 10 | 0.93 | 0.94 | **0.96** |
| | 100 | 0.98 | 0.98 | **0.99** |

| Prediction Type | Text |
|---|---|
| Correct | Poecilia sphenops poecilia sphenops is a species of fish of the genus poecilia known under [..] |
| | The collared sunbird ( hedydipna collaris ) is a sunbird. The sunbirds are a group [..] |
| | The grass bagworm ( eurukuttarus confederata ) is a species of bagworm that only eats grass. [..] |
| Incorrect | Eupithecia parcirufa is a moth in the geometridae family. It is found in bolivia [..] |
| | Nebria kurosawai is a species of ground beetle in the nebriinae subfamily that is endemic to japan [..] |

Table 4: *Left hand side*: Comparison of transfer learning when adding the class "animal" to a model trained to predict 13 other topics. *Right hand side*: A few correct and incorrect zero-shot predictions made by TARS for the target class "animal" (incorrect examples are both classified as "plant").

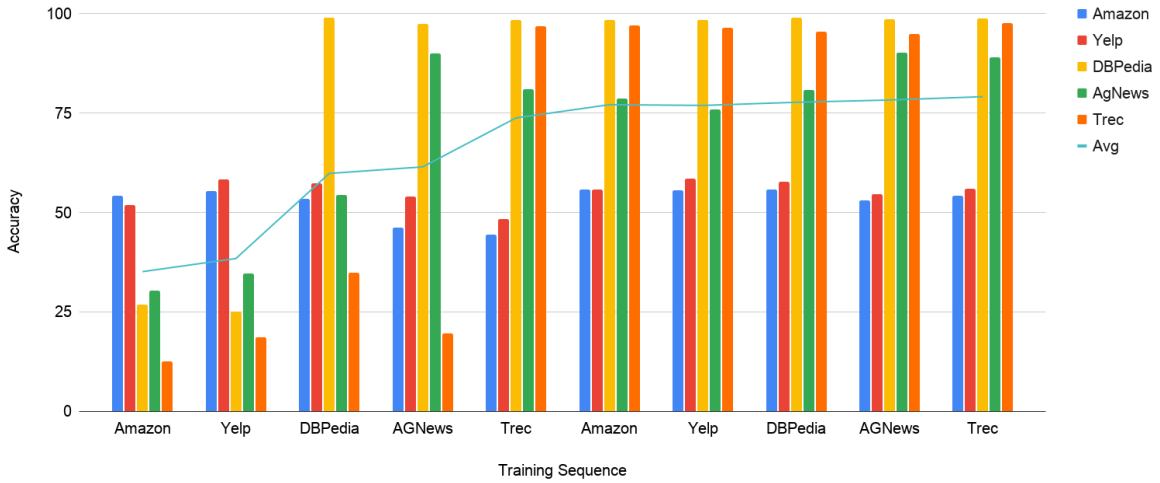


Figure 3: Observed accuracy when a single TARS model is trained on multiple datasets one by one. X axis shows the order of datasets used for training. Y axis depicts the accuracy - bars for individual datasets, trend line for the average across all datasets. The model is capable of retaining the knowledge, and it performs well on *all* the datasets.

as "sunbird". However, this correspondence is not perfect since examples like "*Eupithecia parcirufa is a moth in the geometridae family*" are incorrectly classified as belong to class "plant".

### 3.3 Knowledge Retention Experiment: Continuing Training on Multiple Datasets

As outlined in Section 2.1, a single TARS model is conceptually able to perform text classification on multiple corpora. We evaluate this capability with an experiment in which we sequentially train the same TARS model over all 5 datasets used in previous sections, while monitoring accuracy across all evaluation splits.

Figure 3 shows the results of this experiment: The sequence of training datasets is depicted along the x axis. As shown, we train a TARS model first on AMAZON-FULL and measure the accuracy of the trained model across all datasets. We then use the trained model obtained from the previous step and continue training it on YELP-FULL, after which we again measure the accuracy of the modified model across all datasets. We continue in a similar manner with all other datasets. After TARS is trained with all 5 datasets, we repeat the sequence again to evaluate knowledge retention. As Figure 3 shows, we observe that TARS does not show catastrophic forgetting, and retains the existing knowledge well when newer datasets are introduced in training. The average accuracy improves monotonically throughout the training process. The final TARS model obtained after the training process achieves superior performance across all the datasets with the *same* set of parameters.

## 4 Related Work

**Transfer learning.** Transferring knowledge from one learned task to another relies on exploiting similarities across tasks *e.g,* for question answering, sentiment analysis, and passage re-ranking among others (Min et al., 2017; Severyn and Moschitti, 2015; Howard and Ruder, 2018; Nogueira and Cho, 2019; Dai and Callan, 2019). Here the focus is mainly on transferring knowledge from self- or unsupervised tasks (language modelling) that can exploit large corpora to supervised tasks (like text classification) for which limited labelled data exists. In this work, we are interested in transferring knowledge from one supervised task to another, with varying amount of semantic task similarity and data distribution similarity. In that respect, a unified text-to-text transformer model (Raffel et al., 2019) which tackles all NLP problems (*e.g.* translation, classification, regression) as sequence-to-sequence problems is the most related to our work. We can cast the text classification in that context as prediction of one token with two possible values (0 and 1).

**Zero/few shot learning.** The problem of *zero shot*, and *few shot* learning has lately been proposed in the context of NLP (Han et al., 2018; Geng et al., 2019) using meta-learning. Model agnostic meta-learning (MAML) has been explored to tackle tasks with disjoint label spaces (Bansal et al., 2019). However, these models are not capable of making zero shot predictions. To the best of our knowledge, the only viable zero shot abilities in NLP has been shown in the literature by using pre-trained GPT-2 model (Puri and Catanzaro, 2019). Similar to this work, it utilizes descriptors of the task at hand in natural language, but there it is formulated as question answering problem, rather than as classification problem as we do.

## 5 Conclusion

In this paper we addressed key shortcomings of the existing transfer learning mechanisms for text classification and proposed a novel formulation that transforms it into a generic binary classification problem. The proposed TARS architecture captures the similarity between an input text and the task label to perform text classification. We performed an extensive set of transfer learning tasks from multiple datasets from different domains, including topic classification and sentiment analysis. We showed that TARS is capable of making zero-shot predictions in multiple text classification tasks, and adapts to a new domain faster than competitive baseline models in few-shot learning settings. The proposed model also generalizes the text classification task to the extent that a single model can perform well on multiple text classification tasks simultaneously.

Our efforts in the future are aimed towards optimizing the computational complexity of TARS. We assume that the specific architecture used here has plausible alternatives: one could use separate transformers for task label and text parts of the input, and cross-attend between the two with a shallow transformer. This way the computational complexity can be significantly reduced by assuming different degrees of task and text independence. In the limit, one could have a task embedding and a text embedding and perform cross-attention only at the embedding level (*i.e.* dot-product between task and text embeddings). Another way of reducing computational complexity is by exploring architectural choices of transformer encoders, *e.g.* using complementary lighter transformer encoders, or applying model pruning. These options are orthogonal to our contributions and are subject of active research. Exploration of these trade-offs, as well as exploring a broader range of tasks, are planned for future work. Since, TARS can encapsulate multiple tasks in a single model, it would also be interesting to investigate on the effect of training sequence of different tasks. We encourage the NLP community to utilize our model in other domains with our open source implementation.

## Acknowledgments

The authors would like to sincerely thank the reviewers for spending their valuable time in reviewing and providing constructive comments. We are also grateful to Ralf Herbrich, Adrien Renahy, Anthony Brew, Matti Lyra, Svetoslava Ande, and Urs Bergmann for their suggestions, and support during the work.

## References


Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Erin L Allwein, Robert E Schapire, and Yoram Singer. 2000. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of machine learning research*, 1(Dec):113–141.

Mohamed Aly. 2005. Survey on multiclass classification methods. *Neural Netw*, 19:1–9.

Trapit Bansal, Rishikesh Jha, and Andrew McCallum. 2019. Learning to few-shot learn across diverse natural language classification tasks. *arXiv preprint arXiv:1911.03863*.

Laura Chiticariu, Yunyao Li, and Frederick Reiss. 2013. Rule-based information extraction is dead! long live rule-based information extraction systems! In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 827–832.

Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 985–988.

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.

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. 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 3895–3904.

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.

Bernd Hollerit, Mark Kröll, and Markus Strohmaier. 2013. Towards linking buyers and sellers: detecting commercial intent on twitter. In *Proceedings of the 22nd international conference on world wide web*, pages 629–632.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Nitin Jindal and Bing Liu. 2007. Review spam detection. In *Proceedings of the 16th international conference on World Wide Web*, pages 1189–1190.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-supervised neural text classification. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 983–992.

Sewon Min, Minjoon Seo, and Hannaneh Hajishirzi. 2017. Question answering through transfer learning from large fine-grained supervision data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 510–517.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Raul Puri and Bryan Catanzaro. 2019. Zero-shot text classification with generative language models. *arXiv*, pages arXiv–1912.

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 preprint arXiv:1910.10683*.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pages 502–518.

Aliaksei Severyn and Alessandro Moschitti. 2015. Unitn: Training deep convolutional neural network for twitter sentiment classification. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 464–469.

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*, pages 5754–5764.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, et al. 2019. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.