# Domain-Adaptive Text Classification with Structured Knowledge from Unlabeled Data

**Tian Li***
Peking University
davidli@pku.edu.cn

**Xiang Chen***
Peking University
caspar@pku.edu.cn

**Zhen Dong**
University of California, Berkeley
zhendong@berkeley.edu

**Weijiang Yu**
Sun Yat-sen University
weijiangyu8@gmail.com

**Yijun Yan**
University of California, Berkeley
bunnyyan@berkeley.edu

**Kurt Keutzer**
University of California, Berkeley
keutzer@berkeley.edu

**Shanghang Zhang**†
Peking University
shanghang@pku.edu.cn

## Abstract

Domain adaptive text classification is a challenging problem for the large-scale pretrained language models because they often require expensive additional labeled data to adapt to new domains. Existing works usually fails to leverage the implicit relationships among words across domains. In this paper, we propose a novel method, called Domain Adaptation with Structured Knowledge (DASK), to enhance domain adaptation by exploiting word-level semantic relationships. DASK first builds a knowledge graph to capture the relationship between pivot terms (domain-independent words) and non-pivot terms in the target domain. Then during training, DASK injects pivot-related knowledge graph information into source domain texts. For the downstream task, these knowledge-injected texts are fed into a BERT variant capable of processing knowledge-injected textual data. Thanks to the knowledge injection, our model learns domain-invariant features for non-pivots according to their relationships with pivots. DASK ensures the pivots to have domain-invariant behaviors by dynamically inferring via the polarity scores of candidate pivots during training with pseudo-labels. We validate DASK on a wide range of cross-domain sentiment classification tasks and observe up to 2.9% absolute performance improvement over baselines for 20 different domain pairs. Code will be made available at https://github.com/hikaru-nara/DASK.

## 1 Introduction

Domain shift is common in many natural language processing (NLP) applications. For example, the word "rechargeable" is much more common in electronics product reviews than in book reviews, while the word "readable" is much more common in book reviews. Existing language models [4, 15] have exhibited outstanding performance in text classification tasks, but they fail to generalize to new domains without *expensive labeling and retraining* (Figure 1). To break out the data constraint, some methods with unlabeled data have been proposed as follows.

*Equal contribution.

†Corresponding author. (E-mail: shanghang@pku.edu.cn)

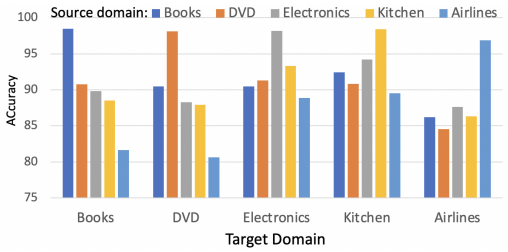



Figure 1: Language models perform worse when domain shift is present. The figure shows the cross-validation results of BERT baseline models trained on 5 domains. The prediction accuracy of the models tested on the trained domain is 5-15% higher than those tested on distant domains.

Existing unsupervised domain adaptation methods for text classification can be grouped into two categories: task-agnostic methods [16, 7, 12, 9] and pivot-based methods. Task-agnostic methods generally ignore the correlation among words across domains, which can contain rich semantic information in an NLP context. In contrast, pivot-based methods use domain-independent words (pivots) to bridge the domain gap by leveraging the correlations between pivots and non-pivots to learn domain-invariant features. Therefore, we would like to marry pivot-based method and pretrained language models to adapt them to novel domains.

The most prominent pivot-based methods are Structure Correspondence Learning (SCL) and its variants [2, 25, 27]. In SCL, the pivots are defined as the words that occur frequently on both source and target domains and behave in similar ways that are discriminable for the classification task[3]. The model can effectively learn domain-invariant features for pivots, but it is more challenging for the non-pivots as they have domain-specific meanings. Therefore a self-supervised auxiliary task is applied to predict the pivots from the non-pivots. As a result, SCL implicitly captures the relationships between words by recognizing co-occurrence patterns between pivots and non-pivots and uses these relationships to infer domain-invariant features for the non-pivots.

However, SCL is limited in that it uses all non-pivots to predict the pivot terms, which leads to a noisy inference problem as very few non-pivots have a real relationship with the pivots. As a result, false correlations often occur for frequently used words such as pronouns (see Figure 2a). Alternatively, Knowledge Graphs (KG) is an effective way to represent complex relationships between concepts in a structured format and do not solely rely on noisy co-occurrence information. Therefore, in this paper, we present a pivot-based domain adaptation method from the KG perspective. Our method, called Domain Adaptation with Structured Knowledge (DASK), follows a 2-step approach as illustrated in Figure 2b. In contrast to SCL, DASK filters out false correlations by building a knowledge graph to explicitly capture the relationships between pivots and non-pivots on the target domain. Then during training, DASK employs a novel knowledge injection (KI) mechanism for the model to learn domain-invariant features of the non-pivots.

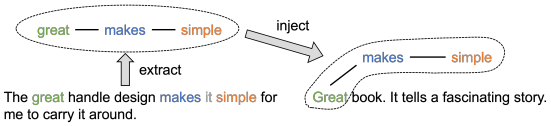
Another critical drawback of SCL is that the pivots are pre-defined only on labeled source domain texts and *unlabeled* target domain texts. There is little to ensure that the pre-defined pivots actually have consistent behavior across domains. To tackle this problem, DASK dynamically learns the pivots during training with pseudo-labels. A memory bank is maintained to keep track of the polarity scores of the candidate pivots on the source domain and the target domain, respectively. The words that have consistently high scores in the memory banks are collected as the pivots at the beginning of each training epoch. As we show, DASK using learned pivots outperforms the static, pre-defined pivots. To summarize, our major contributions are as follows:

- We propose DASK for text classification, which injects knowledge graph facts to better leverage the relationship between words for adaptation across domains.
- We construct a novel knowledge graph with attention scores from BERT to explicitly capture the relationship between pivots and non-pivots on the target domain.
- We design the pivot-induced cross-domain knowledge injection mechanism to learn domain-invariant features for non-pivots using their relationship with pivots.
- We propose to maintain memory banks to learn domain-invariant pivots.

[3]In the existing literature, the definition of pivots often uses the term "behavior". To put it simply, the pivots should be highly correlated to one of the labels across domains.

- We evaluate our method on the task of cross-domain sentiment classification for 20 domain pairs, where our method outperforms strong baselines by up to 2.9%.


(a) Structure Correspondence Learning.




(b) Two-step approach of DASK: extract and inject

Figure 2: a) The example shows a pair of texts from the source domain (top) and target domain (bottom) respectively. Due to the frequency of "it" co-occurring with "great", the model tends to capture this false correlation. b) In contrast, DASK extracts a fact, represented by a triplet (great, make, simple), from the target domain text to filter false correlations. We inject the target domain fact into the source domain text.

## 2 Related Work

### 2.1 Cross-Domain Text Classification

Cross-domain text classification [9, 7, 27, 6] is a fundamental problem in domain adaptation for NLP. Before applying domain adaptation methods to complex problems, like machine translation and question answering, it was extensively researched as a simpler problem. Among text classification tasks, sentiment classification is most representative and established in terms of domain adaptation because there are standard datasets and evaluation protocols [1, 27]. Therefore, we use sentiment classification to evaluate our method. It is important to note that our method can be easily generalized to other tasks such as topic labeling, news classification and so forth.

Methods for cross-domain text classification can be roughly categorized into two classes: task-agnostic methods, and pivot-based methods. The former includes divergence minimization [16, 18, 10], stacked denoising auto-encoders [9], discriminative adversarial training [7, 20], instance reweighting [3] and so forth. There are also some works that combine task-agnostic methods with NLP-specific approaches or models [8, 6]. In contrast, pivot-based methods are different from them in two aspects: 1) they make use of the relationship between pivots and non-pivots to help learn discriminative feature, 2) they focus on learning word-level features instead of instance-level features, which align with the nature of language. These two points make pivot-based methods stand out on NLP tasks.

### 2.2 Knowledge Graph

**Knowledge Graph Construction.** KG can be constructed in a supervised manner, such as DBPedia [11] and Wikidata [21]; semi-supervised manner such as Google's Knowledge Vault [5]; or unsupervised manner, such as MAMA [22]. Among them, we are most interested in MAMA because it uses learned knowledge stored in pre-trained language models without human supervision to construct a KG. With a forward pass of BERT, it collects the words or phrases in the input that have high attention scores with each other to form a candidate fact, which is represented by a triplet *(head, relation, tail)*. Hence it is an easy fit for our knowledge-injected language model P-BERT. We can initialize P-BERT with the same weights from BERT used in MAMA, so that it becomes easier for P-BERT to interpret the knowledge graph facts that are constructed by MAMA.

**Knowledge-Injected Transformers.** Since the emergence of pre-trained language models like BERT [4, 15], many works have followed up to incorporate external knowledge into them [23, 26, 24, 14]. Typically, these works either fuse textual features and factual features at the end of the corresponding encoders to enhance language understanding [24, 26], or use adapters for injection of multiple knowledge sources with the original model fixed [23]. Although most of them tried to utilize open-domain knowledge graphs or knowledge bases, K-BERT [14] takes domain-specific

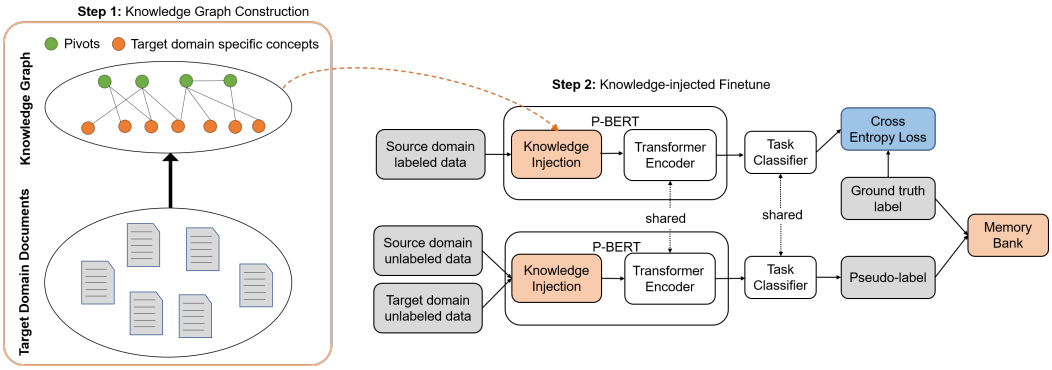



Figure 3: Illustration of DASK. DASK consists of two steps. In step 1 we construct a knowledge graph from target domain unlabeled data. In step 2 we finetune the model on knowledge-injected data and learn the pivots with memory bank.

knowledge as input along with text, and thus is particularly suitable for our task of domain adaptation. Also, by using a flattened sentence tree and the visible matrix, it is compatible to BERT and free from additional pre-training. Therefore, to promote domain transfer with pivots, we develop K-BERT into P-BERT by replacing its KI module with our Pivot-induced Cross-domain Knowledge Injection.

Beyond knowledge-injected transformers, another line of related works is tree-based transformers for processing tree-structured data in general [17, 19]. Despite their power in learning from arbitrary tree-structured data, they are not directly compatible to BERT.

## 3 Method

DASK aims to facilitate domain adaptation with structured knowledge from the target domain. To structure the knowledge in the target domain, we build a knowledge graph to capture the relationship among pivots and non-pivots and inject them into source domain data. This is helpful because the model is able to learn domain-invariant feature for non-pivots (domain-specific) by inferring from their relationship with domain-shared pivots. Specifically, DASK follows a 2-step approach as follows (see Figure 3):

**Step 1.** We construct a knowledge graph (KG) from the target domain texts to model the relationship between pivots and non-pivots. Specifically, it involves 1) extracting candidate facts from sentences, 2) filtering low-confidence facts. Section 3.1 shows details of how we construct the KG.

**Step 2.** After constructing the knowledge graph, we start knowledge-injected finetuning on the source and target domains jointly. At each training step, we obtain labeled texts from the source domain, unlabeled texts from the source domain and target domain. The inputs are fed into P-BERT where all of them are injected with pivot-relevant facts from the knowledge graph (section 3.2), and then forwarded to the transformer encoder and the linear classifier sequentially. Finally, the predictions of labeled data are used to compute the cross-entropy loss with the ground truth labels. Meanwhile, the predictions of unlabeled data are leveraged to generate pseudo-labels for pivot learning (section 3.3). The pseudo-labels of unlabeled data and ground truth labels of labeled data give updates to the polarity scores in the memory banks. Note that during inference the memory bank is not updated.

### 3.1 Knowledge Graph Construction

Prior to training, we build a KG on the target domain corpus to capture the relationship between pivots and non-pivots.

**Candidate Fact Extraction.** We extract candidate facts on a sentence level. An input text is decomposed to a list of sentences. We feed each sentence into BERT to get the attention matrix in

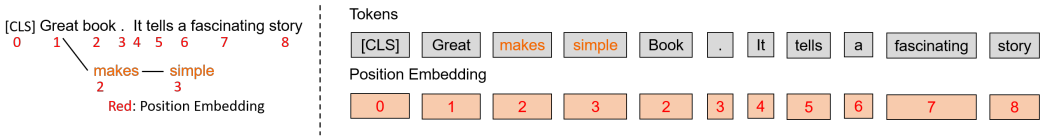



Figure 4: Continuing the example in Figure 2b, we inject the fact triplet *(great, makes, simple)* to the main sentence forming a tree structure (left). On the right, we flatten the tree into a sequence and use the depth of the tokens in the tree as their position embedding index. The highlighted words in orange are the injected fact.

the last transformer encoder layer. As a pre-processing step, the multi-head attention is averaged into single-head so that one pair of words only corresponds to one scalar attention value (if a word consists of multiple tokens, the attentions of the tokens are also averaged). Denote the pre-processed attention matrix as $M$. For each pivot $p$ in the sentence, we search for the words $w_1, w_2$ that have the highest and second highest attention with $p$. Then the fact is formed as the triplet of $w_1, w_2, p$ in their *original order* in the sentence. In addition, each fact is assigned a confidence score $M[p][w_1] + M[p][w_2]$.

**Filtering.** The candidate facts are filtered according to their confidence scores. Those whose confidence scores are under a threshold are removed from the knowledge graph.

Note that although we learn pivots dynamically during training, those pivots all come from a large pivot pool that is constructed before training (see section 3.3). We construct the knowledge graph according to the pivot pool so that we do not have to update the knowledge graph as we learn new pivots and eliminate old ones. In the appendix, we showcase some qualitative results of a KG that we constructed in this way and compare it with the ConceptNet subgraph.

### 3.2 Knowledge Injection

In order to utilize the rich semantics in the KG to learn domain-invariant features for the non-pivots, we propose Pivot-induced Cross-domain Knowledge Injection (PCKI) to inject knowledge facts into input texts. On the other hand, we would like to use a pretrained language model as the feature extractor, so we borrow ideas from K-BERT [14] to let the language model understand the structure of knowledge injected input. As a whole, our model P-BERT is composed of the knowledge injection module and the subsequent transformer encoder, as illustrated in Figure 3.

In the knowledge injection module, for each pivot in the input text, we search for the facts relevant to it and inject them into the text, forming a tree structure. Figure 2b shows an example for a knowledge injected text. In the example, "Great" is a pivot in the sentence on the right, and we have the fact triplet "(great, makes, simple)" in the KG that we extract from the sentence on the left. The triplet is then appended to "great" as a branch. By injecting facts extracted from the target domain into labeled source domain texts, we embed the target domain non-pivots in a labeled context, conditioning on the related pivots. Therefore, our transformer encoder is able to infer their features according to their relationships with pivots, under the supervision of source domain labels.

To feed the knowledge-injected text to the transformer encoder while keeping the structure information, we flatten it into a token sequence, and use position embedding to recover its structure [14]. Specifically, as illustrated in Figure 4, the injected words are inserted before or after the corresponding pivot following their order in the triplet so as to get the flattened token sequence. Meanwhile, to recover the tree structure, we assign the index for position embedding of each token as its depth in the tree. In the example, "makes" and "book" are assigned the same position embedding index 2. In this way, the token sequence is fed into the transformers, with the position embedding informing its structure. Besides, a visible matrix is applied to the self-attention modules in the transformers to control knowledge noise. We refer readers to K-BERT [14] for more details on visible matrix.

### 3.3 Pivot Learning

As a pivot-based method, DASK heavily relies on the domain-shared pivots. In this section, we describe how we select and learn the pivots with memory banks.

Recall that pivots are defined as the words that behave similarly in the source and target domains. Following previous works, we represent the behavior of a word with the labels of the texts in which it appears. The principle is that the label distribution of a pivot should be *low-entropy* (biased towards one label), and *consistent* across domains. While it is applicable to most classification tasks, for the binary sentiment classification task we focus on, we define a polarity score $p(w, D)$ to measure more easily how much the label distribution of $w$ is biased towards the labels on domain $D$:

$$p(w, D) = \frac{|\{l = 1 | l \in b(w, D)\}| - |\{l = -1 | l \in b(w, D)\}|}{|b(w, D)|} \tag{1}$$

where $b(w, D)$ is the label set of $w$ on $D$. And thus the behaviour on the domain pair $(S, T)$ can be characterized as:

$$\bar{p}(w, (S, T)) = \frac{|p(w, S) + p(w, T)|}{2} \tag{2}$$

Taking the absolute average ensures that a word of high score must be biased towards the same label on both domains.

Prior to training, we collect a large pool of candidate pivots from the vocabulary which contains words that appear frequently on both domains. Other words are not taken into account because the precondition is that pivots should be frequent on both domains. To track the polarity scores of the candidate pivots, we build memory banks for the source domain and the target domain respectively. Both of them are initialized by scores computed with equation 1 constrained on source domain labeled data. Then we compute the absolute mean polarity score of the candidate pivots following equation 2 and take the $K$ candidate pivots with the highest score as the initial pivots.

During training, at each training step, we acquire pseudo-labels for the unlabeled texts if the prediction confidence in the softmax logits is over a threshold. The pseudo-labels of the unlabeled source domain inputs, and the ground truth labels of the labeled source domain inputs are used to update the source memory bank, while the pseudo-labels of the target domain inputs are used to update the target memory bank. The update is carried out in a temporal difference style: For each candidate pivot, if it is in the text from domain $D$ labeled as $l \in \{1, -1\}$, then

$$p(w, D) \leftarrow \alpha \cdot p(w, D) + (1 - \alpha) \cdot l \tag{3}$$

where $\alpha$ is the update rate. In this way, we manage to estimate the behavior of words more accurately on the domains, so that domain-invariant pivots are acquired. At the beginning of each epoch, we compute the absolute mean scores of the candidate pivots following equation 2 and take the top-$K$ candidate pivots as learnt pivots. The pivots are kept fixed in the middle of an epoch to avoid overly frequent pivot changes.

## 4 Experiments

To evaluate the efficacy of our proposed method, we extensively experiment on two cross-domain sentiment classification datasets. We compare our method with multiple baselines on 20 domain pairs and provide justification for the improvement of our method. Moreover, we carry out ablation study to respectively assess the effect of knowledge injection and dynamic pivot learning.

### 4.1 Experiment Settings

**Datasets.** We use the standard Amazon-product-review[4] dataset [1]. It contains four types of product reviews: books(B), dvd(D), electronics(E), kitchen(K), which form the four domains in the dataset. Besides, following PBLM [27], we also introduce the Airlines dataset[5].

We evaluate our method on all 20 domain pairs involving the five domains. Each time before training, we randomly sample 400 labeled source domain data for dev set, and the rest 1600 along with all unlabeled data from source and target domain are used for training. During testing on the 2000 target labeled data, we stop the memory bank update, and use the pivots learnt at the best training epoch.

[4]Dataset can be found at http://www.cs.jhu.edu/~mdredze/datasets/sentiment/index2.html

[5]Dataset and process procedures can be found at https://github.com/quankiquanki/skytrax-reviews-dataset

| S→ T | BERT | | | | | |
|---|---|---|---|---|---|---|
| | Base | HATN | DANN | DAAT | DASK | DASK+SCL |
| B→ E | 90.50 | 87.21 | 91.67 | 89.57 | 91.95 | **92.30** |
| B→ D | 90.45 | 89.36 | 89.93 | 89.70 | 90.55 | **90.90** |
| B→ K | 92.46 | 89.41 | 92.80 | 90.75 | **92.85** | 92.75 |
| E→ B | 89.85 | 87.10 | 89.19 | 88.91 | 89.70 | **90.00** |
| E→ D | 88.30 | 88.81 | 88.49 | **90.13** | 88.65 | 89.20 |
| E→ K | 94.20 | 92.01 | 94.54 | 93.18 | 94.35 | **94.65** |
| D→ B | 90.75 | 89.81 | 91.37 | 90.86 | 91.20 | **91.85** |
| D→ E | 91.30 | 86.99 | 91.52 | 89.30 | 88.70 | **92.40** |
| D→ K | 90.85 | 87.59 | 92.16 | 90.50 | 91.80 | **92.35** |
| K→ B | 88.50 | 89.36 | 89.38 | 87.98 | **90.15** | 89.75 |
| K→ E | 93.34 | 90.31 | 93.15 | 91.72 | 92.80 | **93.35** |
| K→ D | 87.90 | 87.89 | 88.89 | 88.81 | 88.40 | **89.45** |
| *Average* | 90.70 | 88.69 | 91.09 | 90.12 | 90.92 | **91.59** |

Table 1: Cross-domain sentiment classification accuracy on 12 domain pairs from Amazon-product-review dataset. Our method is able to outperform all the strong baselines on all domain pairs with the only exception of E→D. For BERT-HATN and BERT-DAAT we use numbers reported by [6].

| S→ T | BERT | | | |
|---|---|---|---|---|
| | Base | DANN | DASK | DASK+SCL |
| A→ B | 81.65 | 81.50 | 82.10 | **84.15** |
| A→ E | 88.85 | **89.53** | 89.35 | 89.10 |
| A→ D | 80.60 | 82.74 | **83.15** | 82.85 |
| A→ K | 89.50 | 89.53 | 89.90 | **90.00** |
| B→ A | 86.18 | 86.66 | 86.30 | **86.70** |
| E→ A | 87.60 | **87.90** | 87.30 | **87.90** |
| D→ A | 84.55 | 86.71 | 84.85 | **86.75** |
| K→ A | 86.30 | 86.56 | 86.50 | **86.80** |
| *Average* | 85.65 | 86.39 | 86.12 | **86.78** |

(a)

| PCKI | SCL | Dynamic | Accuracy | | | | |
|---|---|---|---|---|---|---|---|
| | | | A→ B | B→E | E→K | K→D | D→A |
| | | | 81.65 | 90.50 | 94.20 | 87.90 | 84.55 |
| | ✓ | | 81.85 | 90.95 | 94.40 | 87.95 | 84.90 |
| ✓ | | | 82.10 | 91.90 | 94.25 | 88.20 | 84.75 |
| ✓ | ✓ | | 82.95 | 92.05 | 94.60 | 88.40 | 86.05 |
| ✓ | ✓ | ✓ | **84.15** | **92.30** | **94.65** | **89.45** | **86.75** |

(b)

Table 2: a) Cross-domain sentiment classification accuracy on the 8 domain pairs between airlines dataset and 4 domains from Amazon-product-review dataset. b) Ablation study on PCKI, SCL and dynamic memory bank. We did experiments on 5 domain pairs.

**Baselines.** We use BERT [4] as the feature extractor. Apart from directly finetuning BERT on the source domain, we compare our method with multiple strong baselines incorporating methods proposed by previous works. The baselines are listed as follows:

- **Base**: BERT trained on the source labeled data and directly test on the target labeled data.
- **HATN**: BERT combined with HATN proposed by [13]. It is a prominent variant of Structure Correspondence Learning (SCL).
- **DANN**: BERT combined with the popular adversarial training method, DANN [7].
- **DAAT**: BERT-DAAT proposed by [6]. It combines target domain post-training along with domain adversarial training to boost domain adaptation.

To show the compatibility of DASK with existing methods, we stack SCL on top of our method. Since there was no existing work applying SCL on transformers, we manufactured a training scheme for SCL that mimics MLM in BERT [4] pretraining, that is, to replace the pivots as [MASK] tokens and ask the model to recover the masked tokens given the context.

**Hyperparameter Tuning.** For all the methods in our experiments, we set the learning rate to 2e-5, warmup 0.1, batch size 32, and select weight decay in {1e-4, 2e-4, 3e-4}. For adversarial training, we select $\gamma$ in the gradient reversal layer from $\{0.15, 0.25, 0.5, 0.75, 1.0\}$. For KG fact filtering, we set the confidence threshold in the interval $[0.1, 0.45]$, according to the confidence distribution. Besides, we leverage the stopwords in the NLTK library and discard the facts that include the stopwords. For SCL, we apply a balance factor $\lambda$ to the pivot-prediction loss, and choose its value from $[0.1, 0.5]$. To make the learning progress smoother, we also use a linear warmup to $\lambda$ and set the warmup rate to $0.1$. In addition, since our sentiment classifier is trained on the source labeled data, and the size of labeled data is 6-20 times smaller than that of unlabeled data, our

model see each labeled data for 6-20 times in one epoch. In order to prevent overfitting, we update the sentiment classifier once in 5-11 training steps. For the dynamic memory bank, we update the pivots every 10 training steps, and the top 500 words are selected as pivots.The learning rate of word sentiment score is set to 1e-4 or 2e-4, and the pseudo-labeling confidence threshold to 0.9.

### 4.2 Experimental Results

Table 1 shows the performance of our method on the 12 domain pairs from Amazon-product-review dataset, compared with multiple baselines.

Among all those methods, we observe that DASK+SCL gives the best performance on average. It outperforms the Base model by 0.89%, and beats the other strong baselines by 0.50%-2.90% on an average basis over all domain pairs. On the other hand, DASK also has competitive performance. It is better than DAAT and HATN by 0.8% and 1.43%, outperforms the Base Model by 0.22%. Although its average accuracy is slightly lower than DANN, it gives the best performance on the B→ K and K→B settings.

Moreover, we conduct experiments on 8 domain pairs involving the Airlines dataset. Table 2a shows the performance of our method on those settings. Although this setting is more difficult than domain pairs within Amazon-product-review dataset, we are still able to outperform Base on all of the domain pairs, by 1.13% on average. This shows that, by exploiting correlation among words, our method can help the model capture relationship between pivots and non-pivots even from distant domains. Besides, our method betters DANN by 0.39%. Although this is not a thick margin, it is non-trivial because 1) DANN itself is a very competitive method, 2) we do not expect to beat DANN by any larger margin, but to propose a strong baseline for domain adaption from a KG perspective, 3) our method is orthogonal to DANN and thus can be coupled with DANN to achieve even better performance.

Note that our Base model performs better than the previous state-of-the-art BERT-DAAT. This owes to hyperparameter tuning where BERT-DAAT uses 1e-2 for weight decay while we use 1e-4. It is not open-sourced so we cannot reproduce their results for better comparison. Nevertheless, it does not mean our performance gain comes from hyperparameter tuning as we analyze in the ablation studies below.

| KI method | KG | Accuracy |
|---|---|---|
| normal | subgraph | 80.45 |
| normal | learnt | 77.45 |
| PCKI | subgraph | 79.50 |
| PCKI | learnt | **82.10** |
| Base | | 81.65 |

Table 3: Ablation studies on knowledge injection mechanism and KG construction method. All experiments are done on the A→B domain pair.

**Ablation Study.** As a sanity check, we performed experiments to study the effect of our choices of KI method and the knowledge graph. We compare PCKI versus normal KI method which inserts knowledge to every word possible, and our learnt KG versus the ConceptNet subgraph. From table 3, it can be observe that none of the configurations works other than PCKI plus learnt KG, our proposed method.

More importantly, to analyze how much each part of our method contribute to the performance gain, we conduct ablation study on three aspects: PCKI, SCL and pivot learning (Dynamic). We did experiments on 5 domain pairs as shown in table 2b. From the results, we have the following observations: 1) PCKI and SCL are both able to help DA independently, and PCKI generally works better than SCL, 2) Jointly applying them can further boost the performance, which implies that they complement each other in the ability of exploiting the correlation among pivots and non-pivots, 3) dynamically learning the pivots improve the performance by a large margin compared to using a set of pre-defined pivots. This supports that the learnt pivots are more beneficial to domain adaptation than the pre-defined ones.

| Entity | ConceptNet Subgraph | Learnt KG |
|---|---|---|
| great | (great, related to, good)<br>*(great, related to, alexander)*<br>(great, similar to, large)<br>(mega, related to, great)<br>(great, related to, awesome)<br>*(rocking, related to, great)*<br>*(lies, related to, great)* | (looks, surprisingly, great)<br>(great, is, awesome)<br>(great, is, excellent)<br>**(great, save, $)**<br>**(really simple, got, great)**<br>**(easy, seems, great)**<br>(also, looks, great) |
| simple | *(simple, related to, unsophisticated)*<br>*(five needled, similar to, simple)*<br>(simpler, form of, simple)<br>*(simple, synonym, unsuspecting)*<br>*(cakewalk, related to, simple)*<br>(easy, related to, simple)<br>(plain, related to, simple) | **(simple, is, amazing)**<br>**(charging, simple, quick)**<br>**(excellent condition, putting, simple)**<br>(plain, and, simple)<br>(the setup, fairly, simple)<br>**(amazon, simple, fast)**<br>(makes, it, simple) |

Table 4: Visualization of the triplets in learnt knowledge graph compared to Conceptnet subgraph. Both graphs are extracted for the domain pair B→E. The **bold** triplets indicate a relationship between non-pivots on the E domain and pivots between B→E domain pair. The *italicized* triplets are knowledge noise that irrelevant to the target domain. The results show that our learnt knowledge graph better models the relationships between pivots and relevant non-pivots and avoids irrelevant knowledge noise.

### 4.3 Qualitative Results

#### 4.3.1 Knowledge Graph

To demonstrate the advantage of the learnt KG over the ConceptNet subgraph in modeling relationship between pivots and non-pivots for domain adaptation, we visualize an example of ConceptNet subgraph and our KG in Table 4. Each cell contains the triplets involving the concept on the left. For example, the top-left grid contains the triplets in ConceptNet subgraph that involves "great", etc. In the table, the ConceptNet subgraph is the one-hop subgraph induced by words on the electronics (E) domain corpus and the learnt KG is the result of applying our method described in 3.1 to the B→E domain pair. Here, we consider the concepts "great" and "simple" because "great" is a pivot that carries positive sentiment, while "simple" is a positive non-pivot on E domain that carries positive sentiment ("simple" could mean "unsophisticated" in book reviews, which is a negative word for stories.). We want to examine how our learnt KG explicitly models the relationship between pivots and non-pivots and therefore benefits domain adaptation with PCKI. From table 4, we notice the following facts:

In both ConceptNet subgraph and our KG, "great" is related to other positive pivots (good, awesome, large, excellent), and some domain-specific concepts. However, in ConceptNet subgraph, the related domain-specific concepts is not necessarily related to electronic products. This is because ConceptNet is a commonsense KG extracted from open-domain corpus such as Wikipedia. Therefore, concepts like "alexander", "rocking", and "lies" that are irrelevant to electronics would introduce noise when used in knowledge injection.

In contrast, our KG only contains concepts that occur in electronics domain corpus, which avoids knowledge noise. Besides, in our KG, "great" is usually related to other sentiment indicating words or phrases, such as "save $", "really simple", "easy" and so forth. Notably, those words or phrases probably does not convey consistent sentiment on source domain (Books), and thus the model cannot learn discriminative features for them without knowledge injection. Coupled with PCKI, these relationships will enable our model to understand those target domain-specific concepts and to learn discriminative features for them.

Similarly, ConceptNet relates "simple" to synonyms like "easy" and various irrelevant domain-specific concepts such as "cakewalk" and "five needled". In contrast, our KG relates it to pivots such as "amazing" and "excellent condition". These connections between pivots and non-pivots are able to help model learn discriminative feature with PCKI.

| S→T | Initial Pivots | Learned Pivots |
|---|---|---|
| B→E | anything, *pages,* **comfortable**, got, wanted, left, completely, *paper,* went, began, pick, seems, **trouble, average,** add, example, stand, fell, effort, self, expectations, virtually, *artist* | **great, best, love,** history, personal, lives, involved, **larger, enjoyed, trust,** eye, also, **thank, definitely,** knowledge, **honest,** means, leader, **critical,** reviewed, draw, **sweet,** drawing |
| E→K | back, tried, minutes, different, year, cut, anything, **work**, *charge,* goes, service, months, several, received, products, days, four, imagine, selling, point, hot, track, happen, something, whole, went, experience, **neither,** ok, pass, half, *image* | **well, works,** use, **good, nice,** used, also, **easily, like, recommend, quality,** hand, need, music, **heavy, beautiful,** far, sound, included, paper, including, **fairly,** taking, watch, operation, home, become, turning, includes, **lot,** transfer, connects |
| K→D | back, money, end, thought, left, trying, second, check, coming, stand, help, alone, times, instead, **huge,** *sheets, months,* forget, *temperature,* count seconds, *rust,* saying, plan, *ingredients,* twice, picture, *loaf,* putting, *delivery, turned, brown,* went, behind, directions, **fair, correct,** elsewhere, sort | **well,** one, **every, like, good,** makes, really, set, also, **recommend, especially,** seen, **ever, beautiful, always,** use, **friend, kind,** etc, *pans,* might, *tea,* want mostly, wife, food, almost, let, version, long, heat, much, handles, glasses, party, mix, duty, green, come |

Table 5: Qualitative results of dynamically learnt pivots at the end of training compared to the initial pivots. To be clear and concise, we do not show their intersection but only show their difference. **Bold** ones are the words we would probably deem as pivots from human instinct. *Italicized* ones are the words that bias to the source domain, i.e. source domain-specific concepts. This visualization demonstrates that the pivots that we learn from dynamic memory banks are more consistent with human sense and more domain-general than those pre-defined by rules in previous works.

Moreover, our KG is more flexible and specific than ConceptNet subgraph. The vast majority of ConceptNet subgraph relations are "related to", "similar to", but the relation in our Bilevel KG has better diversity. Conventionally, the relation set of a KG is designed by human experts and is thus relatively limited. In contrast, we extract Bilevel KG with a language model from the domain corpus. Although some of the extracted relations are not verb, they actually indicate anonymous relationship with complex semantic meaning that should not be plausibly marked with "related to". For example, (great, save, $) actually indicates anonymous relationship between "great" and "save $". Those anonymous relationships greatly enrich the content of the knowledge graph and avoids using plausible terms like "related to" which could induce knowledge noise.

In conclusion, the ConceptNet subgraph is noisier than the learnt KG. The anonymous relationship in the learnt KG taps the potential of an NN language model to interpret it.

### 4.4 Dynamically Learned Pivots

To demonstrate the effect of pivot learning, we visualize the initial pivots and our learnt pivots on B→E, E→K and K→D domain pairs in table 5. The "initial pivots" are the pivots from the initialized memory banks before training. Note that they are identical to the pre-defined pivots used by previous works [27]. The "learnt pivots" are the pivots from the memory banks at the end of training (see section 3.3). We evaluate the quality of a set of pivots by two criteria. Firstly, we expect the pivots are sentiment indicating, which means that we expect the pivots to have significant sentiment polarities. Secondly, to promote the domain transfer, we aim to prevent the pivots from being biased to the source domain. We report the difference of the two sets of pivots in table 5 *without cherry picking*. From table 5, we have the following observations:

- **Bold** words are some typical pivots that we judge from human instinct. The number of this kind of words, such as "great", "best" and "love", is greatly increased by dynamically learning the pivots with the memory bank.
- *Italicized* words are some source domain-specific concepts which are biased to the source domain and may degrade the performance in the target domain. We observe less source domain-specific concepts after pivot learning. This shows that our method is able to learn domain-general pivots in order to better promote the domain transfer.

These observations owe to our dynamic memory banks, which maintain the polarity scores of the common words between the two domains. When computing the initial polarity scores of the words, only labeled data on the source domain is available, which makes the induced pivots prone to bias to the source domain. In contrast, our memory bank is dynamically updated during training not only with pseudo-labels of the unlabeled text on both source and target domains, but also with the ground-

truth labels of the labeled text. Therefore, our memory banks obtain comprehensive understandings of the words during training, and thus the learnt pivots will be more accurate and less biased to the source domain.

## 5 Conclusion

In this paper, we proposed Domain-Adaptive text classification with Structured Knowledge (DASK) which elegantly marries pivot-based methods, knowledge graph and pretrained language models. It builds a knowledge graph from the target domain data to model the relationship between pivots and non-pivots. During training, DASK injects knowledge into the input sentence by means of pivot-induced cross-domain knowledge injection, and P-BERT is proposed to encode the knowledge-injected sentence. In this way, DASK learns domain invariant features by capturing the relationship between pivots and target domain-specific non-pivots. We conducted experiments on cross-domain sentiment classification and showed that DASK outperformed strong baselines in prediction accuracy by up to 2.9% on 20 domain pairs. Despite that our experiments are focused on sentiment classification, DASK generalizes to other cross-domain text classification tasks, as long as pivots are well-defined for the task. We reserve it as future work to extend DASK to other tasks.

## References


[1] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 2007.

[2] John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, 2006.

[3] Xiang Chen, Yue Cao, and Xiaojun Wan. Wind: Weighting instances differentially for model-agnostic domain adaptation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2366–2376, 2021.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *SIGKDD*, 2014.

[6] Chunning Du, Haifeng Sun, Jingyu Wang, Qi Qi, and Jianxin Liao. Adversarial and domain-aware bert for cross-domain sentiment analysis. In *ACL*, 2020.

[7] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 2016.

[8] Deepanway Ghosal, Devamanyu Hazarika, Abhinaba Roy, Navonil Majumder, Rada Mihalcea, and Soujanya Poria. Kingdom: Knowledge-guided domain adaptation for sentiment analysis. In *ACL*, 2020.

[9] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*, 2011.

[10] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. Adaptive semi-supervised learning for cross-domain sentiment classification. *arXiv preprint arXiv:1809.00530*, 2018.

[11] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 2015.

[12] Tian Li, Xiang Chen, Shanghang Zhang, Zhen Dong, and Kurt Keutzer. Cross-domain sentiment classification with contrastive learning and mutual information maximization. In *ICASSP*, 2021.

[13] Zheng Li, Ying Wei, Yu Zhang, and Qiang Yang. Hierarchical attention transfer network for cross-domain sentiment classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[14] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. K-bert: Enabling language representation with knowledge graph. In *Proc. of AAAI*, 2020.

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

[16] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015.

[17] Vighnesh Leonardo Shiv and Chris Quirk. Novel positional encodings to enable tree-based transformers. In *NeurIPS*, 2019.

[18] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, 2016.

[19] Zeyu Sun, Qihao Zhu, Yingfei Xiong, Yican Sun, Lili Mou, and Lu Zhang. Treegen: A tree-based transformer architecture for code generation. In *AAAI*, 2020.

[20] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017.

[21] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 2014.

[22] Chenguang Wang, Xiao Liu, and Dawn Song. Language models are open knowledge graphs. *arXiv preprint arXiv:2010.11967*, 2020.

[23] Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Cuihong Cao, Daxin Jiang, Ming Zhou, et al. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*, 2020.

[24] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. Kepler: A unified model for knowledge embedding and pre-trained language representation. *ACL*, 2021.

[25] Jianfei Yu and Jing Jiang. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In *EMNLP*, 2016.

[26] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*, 2019.

[27] Yftah Ziser and Roi Reichart. Pivot based language modeling for improved neural domain adaptation. In *Proc. of ACL*, 2018.