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

Weijiang Yu
Sun Y at-sen University
weijiangyu8@gmail.com

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

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

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

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

2022. 9. 25 • ChongQing

—— IJCAI 2022

https://github.com/hikaru-nara/DASK

gesis Leibniz-Institut für Sozialwissenschaften

Forschungszentrum · Research Center L3S

Reported by JiaWei Cheng

Introduction

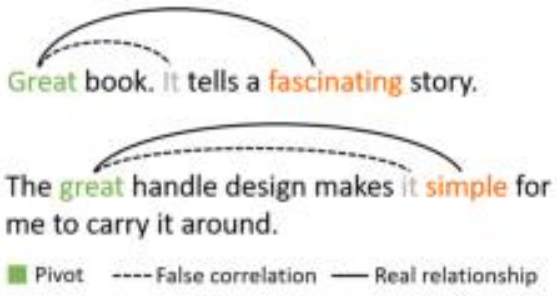
Existing language models have exhibited outstanding performance in text classification tasks, but they fail to generalize to new domains without expensive labeling and retraining .

task-agnostic methods

pivot-based methods

Structure Correspondence Learning (SCL)

Introduction



(a) Structure Correspondence Learning.

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

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.

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.

Overview

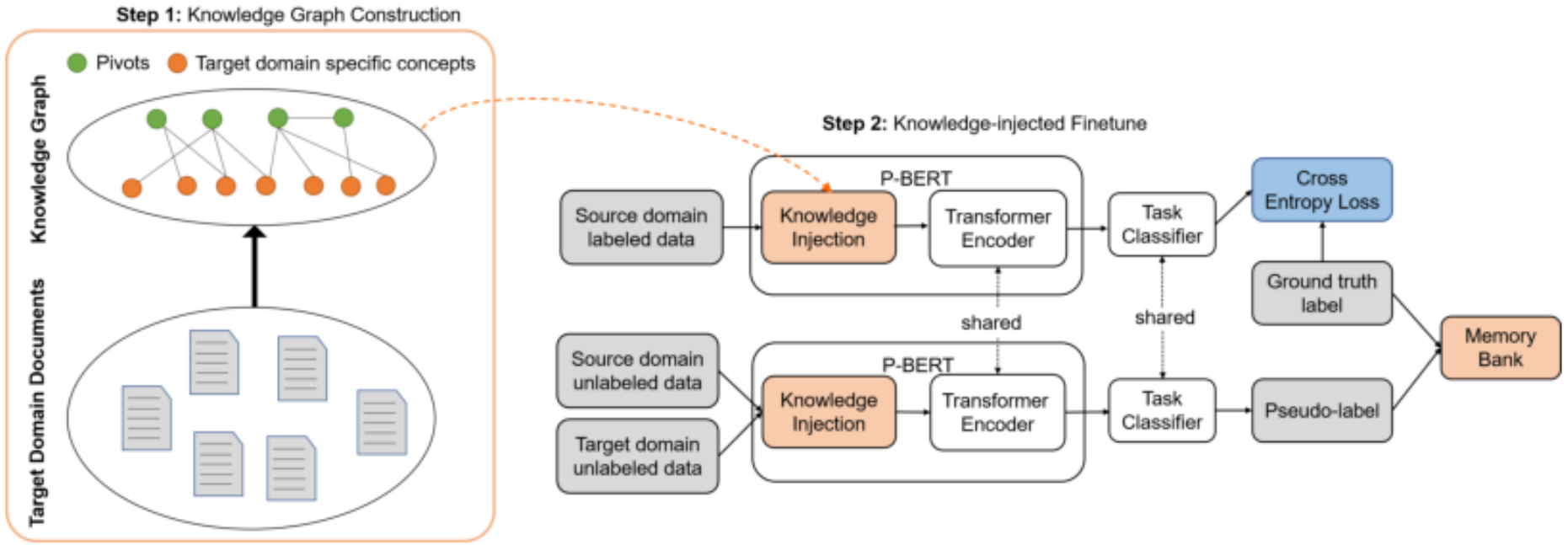
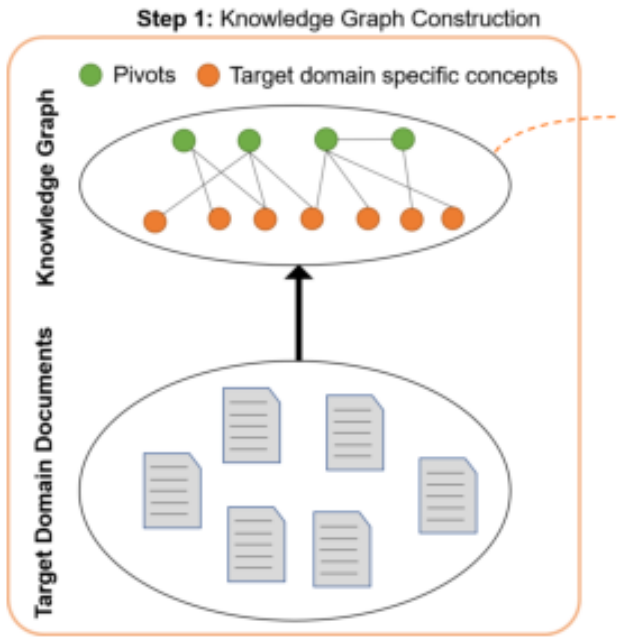


Figure 3: Illustration of DASK. DASK consists of two steps. In step 1 we construct a knowledge graph from target domain unlabeled data. In step 2 we finetune the model on knowledge-injected data and learn the pivots with memory bank.

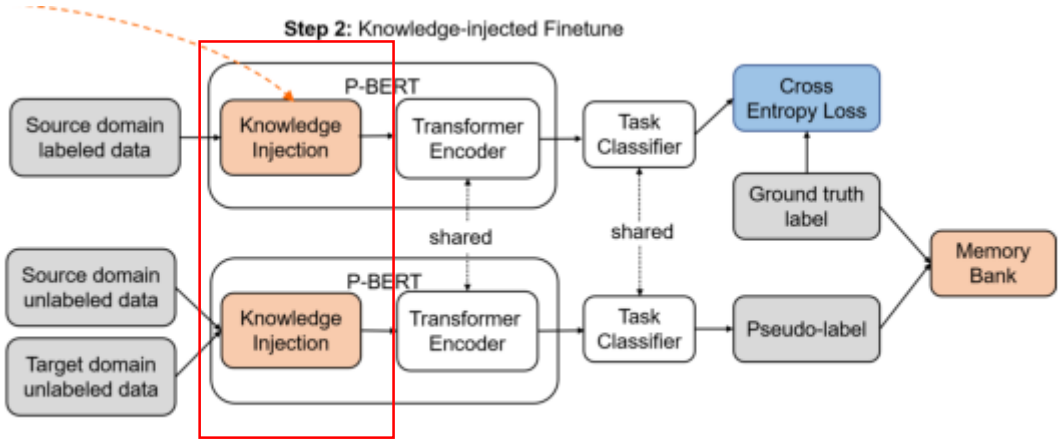
Method



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 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.

Method



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

b):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.

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.

Method

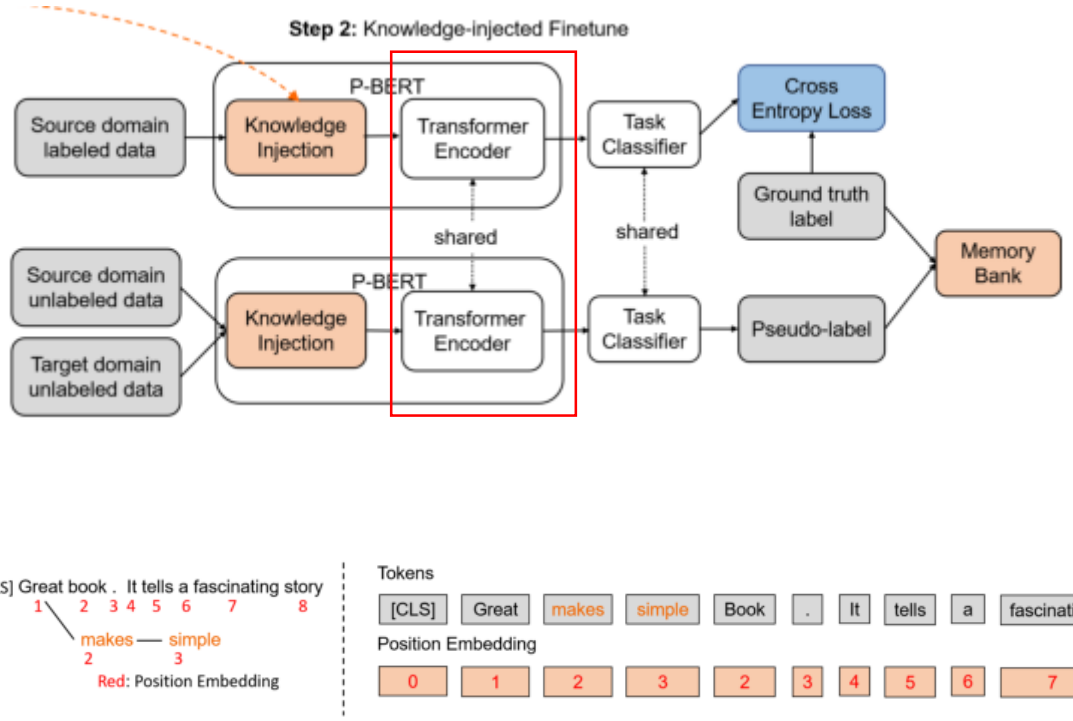
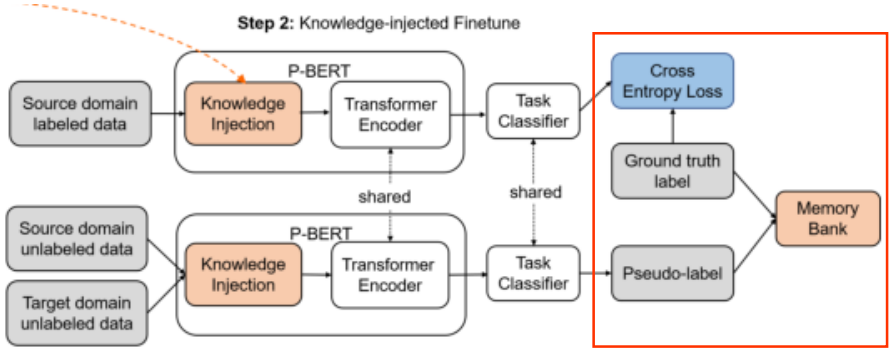


Figure 4: Continuing the example in Figure 2b, we inject the fact triplet *(great, makes, simple)* to the main sentence forming a tree structure (left). On the right, we flatten the tree into a sequence and use the depth of the tokens in the tree as their position embedding index. The highlighted words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

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

Method



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:

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

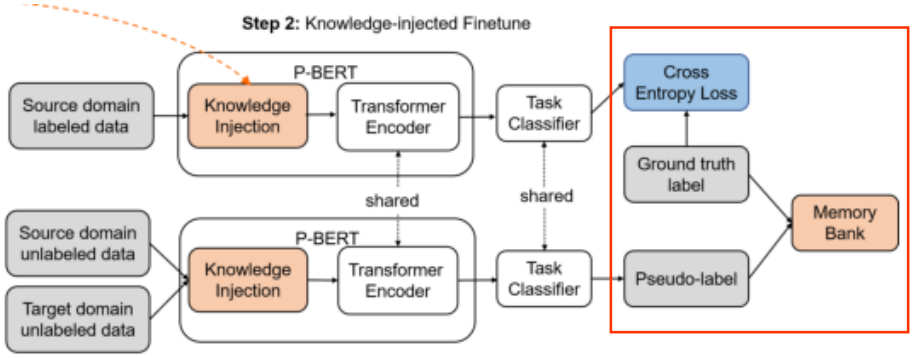
where $b(w, D)$ is the label set of w on D. And thus the behaviour on the domain pair (S, T) can be characterized as:

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

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

Method

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



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

where α is the update rate.

Experiments

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].

Experiments

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.

Experiments

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.

Experiments

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

Experiments

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.

Thanks!