Using uncertainty for multi-domain text classification

Kourosh Meshgi, Maryam Sadat Mirzaei
[1]RIKEN Center for Advanced Intelligence Project (AIP), Japan
https://doi.org/10.36505/ExLing-2020/11/0033/000448

Abstract

Multi-domain learning allows for joint feature detection to promote the performance on a learning task. The shared feature space, however, has limited capacity and should include only the most discriminative task-independent features that are useful for all the tasks. To this end, we proposed a global-local task uncertainty measure to monitor the usefulness of features for all tasks, increasing their effectiveness and generalizability while disentangling them from task-specific features that are not helpful for other tasks. Besides, this measure can utilize unlabeled domain data, tapping the vast reserves of unlabeled data to have even better features. An experiment on a multi-domain text classification shows that the proposed method consistently improves the baseline's performance and improves the knowledge transfer of learned features to unseen data.

Keywords: Multi-Domain Learning, Uncertainty, Feature Disentanglement

Introduction

There are numerous applications for text classification, ranging from document categorization to author identification and sentiment analysis. While different texts may include different vocabulary, grammar, writing styles, and collocations, there are some intrinsic commonalities and potential similarities among them that make multitask learning (MTL) a suitable tool to tackle this problem. MTL exploits such commonalities and tries to learn a shared feature space that improves the accuracy and generalization of all text domains simultaneously. To do this, MTL shares some learned features from one task to another that might need it but has no dedicated data to learn it. It also focuses on learning only relevant features that can be useful for all tasks. It uses other tasks as a regularizer for one, that empowers the tasks to tackle noisy or high dimensional data. It also shifts the features toward generalization since the emergent features should be useful for other tasks and domains (Ruder, 2017). However, if a classifier only employs generic features, its performance on the target task would be suboptimal (Liu, Qui, & Huang, 2017). On the other hand, fine-tuning each task diminishes others' performance or restricts the generalization of the shared features (Lee et al., 2017). To this end, the idea of shared-private MTL has been proposed in which, along with the shared feature space, each task has its own private feature space that can learn task-specific features and boost the task performance (Liu et al., 2017).

However, some task-specific features still emerge in the shared feature space that is not generalizable (thus, not useful for all tasks), wasting the capacity of the shared space to learn more desired features (Ganin & Lempitsky, 2015). It is potentially redundant to the features emerging in the private space of some tasks. To maximally improve the generalization of the shared feature space, the task-specific and task-independent features should be separated (i.e., disentangled), and feature redundancy should be omitted.

To this end, we propose the notion of global-local task uncertainty discrepancy, in which we monitor how private-only and shared-only features can classify a specific domain, measure the uncertainty of the labels given by each feature sets, compare them, and maximize the difference between the uncertainty of two classifiers (built by shared or private-only features) for each task. In another view, the MTL loss function is considering the label inaccuracy for each task and the amount of task uncertainty caused by shared features for each task (that can be explained with some other features emerging in private feature space). This will disentangle shared and private feature spaces, pull the useful features from private space to the shared, and expell the redundant ones.

Disentangling features using uncertainty

In a deep multitask network, several tasks $k=1,\ldots,K$ are jointly trained on their respective data D_k, to form a shared parameter space in which shared features $\boldsymbol{f}_S$ emerges. If all tasks only use $\boldsymbol{f}_S$ as the feature set, the setting is called fully-shared MTL. However, fully-shared settings are not capable of capturing the complicated relationship between tasks, and therefore, the emergent features have suboptimal performance. To this end, private-shared architecture allocates a private feature space $\boldsymbol{f}_P^{(k)}$ to each task k. In collaboration with shared features, private features capture more patterns emerging in each task, which promotes the performance of individual tasks. However, in this naïve form, there is no guarantee that the useful and generalizable features emerge in the shared space. Also, some features are better suited for one or a few tasks (task-specific features) that can emerge in shared feature space-wasting the capacity of this space, limiting the overall effectiveness of shared features and jeopardize the generalizability of these features to unseen tasks. Thus, the task-specific features should be forced to evolve only in the private space of tasks, while task-independent features should be encouraged to appear in shared space.

We propose a global-local uncertainty measure to see if a feature is useful for all tasks, or only a few. Instead of directly using training or development errors that are suboptimal indicators for learning progress (Kendall, Gal, and Cipolla 2018), we used task uncertainty to have a sense of how the MTL is affected by changing the shared feature space in a particular way. This change is typically a result of supervised training of task k, where the network parameters (including the corresponding private space and the shared space) are tuned to reduce the training loss of the task. Here, we calculate the uncertainty of each task using

only shared features (disabling private features) on its dev set, when a task changes the shared feature space. We accumulate these values to provide the global task uncertainty measure, L_{gu}. We also include the uncertainty of the current task (i.e., the task that is being trained) using both shared and private features as a local task uncertainty loss, L_{lu}. These uncertainty measure are added to the task loss, L_{task}, to provide the final loss for task k.

$$L^{(k)} = L^{(k)}_{task} + \lambda_1 L_{gu} + \lambda_2 L^{(k)}_{lu}$$

in which λ_1 and λ_2 are regularization coefficients, and L_{task} is the sum of loss of each task. Note that instead of dev set, L_{gu} can be calculated using unlabeled data as uncertainty calculation doesn't require label.

Experiments and results

To measure the effectiveness of the proposed uncertainty measure on feature disentanglement and on the performance of the classifier, we conduct two sets of experiments. In the first experiment (Table 1), we train our baseline text classifier on 16 different text domains and compare it with a single-task classifier and other MTL classifiers, and investigate the effect of proposed uncertainty (UPS-MTL) on the performance. We also used unlabeled data in UPS-MTL+ to demonstrate the power of including unlabeled data.

Table 1. The accuracy (%) of the model trained on 16 tasks, compared to its single task LSTM baseline and other MTL classifiers. (first, second, third)

DOMAIN	LSTM	MT-CNN	FS-MTL	SP-MTL	ASP-MTL	UPS-MTL	UPS-MTL+
BOOKS	79.5	84.5	82.5	81.2	84.0	86.3	86.8
ELECTRONICS	80.5	83.2	85.7	84.7	86.8	88.4	88.9
DVD	81.7	84.0	83.5	84.0	85.5	87.0	87.5
KITCHEN	78.0	83.2	86.0	85.2	86.2	89.3	89.6
APPAREL	83.2	83.7	84.5	86.5	87.0	87.8	88.1
CAMERA	85.2	86.0	86.5	88.0	89.2	89.4	89.9
HEALTH	84.5	87.2	88.0	87.2	88.2	90.1	90.5
MUSIC	76.7	83.7	81.2	83.0	82.5	84.4	84.9
TOYS	83.2	89.2	84.5	85.2	88.0	88.2	88.7
VIDEO	81.5	81.5	83.7	83.2	84.5	87.0	87.4
BABY	84.7	87.7	88.0	86.7	88.2	90.7	91.0
MAGAZINES	89.2	87.7	92.5	92.0	92.2	92.7	93.3
SOFTWARE	84.7	86.5	86.2	87.0	87.2	87.5	87.8
SPORTS	81.7	84.0	85.5	87.2	85.7	86.3	86.7
IMDB	81.7	86.2	82.5	84.7	85.5	85.7	86.3
MR	72.7	74.5	74.7	76.0	76.7	76.8	77.1

Table 2. The accuracy of the model trained on 15 domains and tested on the leave-one-out domain.

LOO DOMAIN	SP-MTL	ASP-MTL	UPS-MTL	LSTM-X
BOOKS	82.2	83.2	86.5	82.9
ELECTRONICS	84.7	82.2	86.3	83.3
DVD	85.2	85.5	86.4	84.0
KITCHEN	85.0	83.7	86.6	82.2
APPAREL	85.2	87.5	87.4	84.6
CAMERA	86.7	88.2	88.3	86.1
HEALTH	85.5	87.7	88.9	86.6
MUSIC	80.0	82.5	86.4	81.2
TOYS	86.2	87.0	87.2	84.3
VIDEO	85.7	85.2	86.7	83.5
BABY	83.5	86.5	86.6	85.4
MAGAZINES	89.5	91.2	91.1	89.8
SOFTWARE	87.0	85.5	87.3	85.6
SPORTS	83.7	86.7	86.9	83.7
IMDB	87.2	87.5	87.6	84.5
MR	74.0	75.2	75.4	74.0

In the second experiment (Table 2), we probe the effect of the proposed regularization on the generalization of the emergent shared features. In this experiment, we train the competing classifiers on 15 domains and test them on the remaining unseen domain. We also examine the viability of transfer learning by putting learned shared features of UPS-MTL in a single task LSTM-based

classifier (denoted by LSTM-X). The dataset includes 14 product review collections and two different movie review repositories for the task of binary classification similar to (Liu et al., 2017). Each category has a train/dev/test/unlabeled data size of approximately 1400/200/400/2000. We compared our method with a single task classifier (LSTM), PS-MTL, and Adversarial version of that APS-MTL proposed in (Liu et al., 2017), fully shared (FS) MTL, and also MT-CNN (Collobert & Weston 2008) with partially shared CNN for different tasks. The baseline classifier is an LSTM (128 units) operating on GloVe embedding of the input, followed by a dense and a Softmax layer. The regularization weight is tuned by cross-validation (λ_1=.01, λ_2=.025).

As Table 1 shows, the proposed regularization improves the performance of its baseline (PS-MTL) in almost all categories, and although not significant (p=*0.06*), it yields better performance compared to applying adversarial training on the same baseline. Adversarial training in APS-MTL aims to push out domain-specific features from shared space, which indirectly leads to better performance. However, in our proposed algorithm (UPS-MTL), the uncertainty regularization pushes out any features that increase global task uncertainty (punishes task-specific features) and, at the same time, reduces local task uncertainty (by encouraging each task to find better task-dependent features in their private space). The task-independent features in a shared layer of our proposed method are more generalizable compared to FS-, PS- and APS-MTL, as it is demonstrated by the second experiment (Table 2). Also, it is evident that the transferring emergent features in our model to a simple classifier without fine-tuning leads in an acceptable result (LSTM-X).

Conclusion

In this paper, we proposed a global/local task uncertainty regularization to disentangle task-independent and task-specific features in a private-shared MTL setting. Findings show that the features obtained by this measure improve the performance of MTL while providing a strong feature basis for transfer learning targeted at unseen domains.

References

Collobert, R.; and Weston, J. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. Proc. ICML'08.

Ganin, Y., & Lempitsky, V. 2015. Unsupervised domain adaptation by backpropagation. Proc. ICML'15, 1180–1189, Lille, France.

Kendall, A.; Gal, Y.; and Cipolla, R. 2018. Multitask learning using uncertainty to weigh losses for scene geometry and semantics. Proc. CVPR'18, USA, 7482–7491.

Lee, S. W., Kim, J. H., Jun, J., Ha, J. W., & Zhang, B. T. 2017. Overcoming catastrophic forgetting by incremental moment matching. Proc. NIPS 2017, 4652-4662, USA.

Liu, P.; Qiu, X.; and Huang, X. 2017. Adversarial multitask learning for text classification. Proc. ACL'17, 1–10, Vancouver, Canada.

Ruder, S. 2017. An overview of multitask learning in deep neural networks. arXiv.