Neural Processing Letters (2022) 54:2901–2918
https://doi.org/10.1007/s11063-022-10746-4

Domain Adaptation for POS Tagging with Contrastive Monotonic Chunk-wise Attention

Rajesh Kumar Mundotiya[1] · **Arpit Mehta**[1] · **Rupjyoti Baruah**[1]

Accepted: 13 January 2022 / Published online: 2 February 2022

Abstract

Part of Speech (POS) tagging is a sequential labelling task and one of the core applications of Natural Language Processing. It has been a challenging problem for the low resource languages. Sequential labelling algorithms aim to model relationships among the words of a sentence. Availability of annotated datasets in ample amounts is another challenge for low resource languages. Contrastive training has been tried as a robust approach that captures the essential features during model training and based on this, Contrastive Monotonic Chunk-wise attention with CNN-GRU-Softmax (CMCCGS) model architecture has been proposed for POS tagging. It learns optimal features in a low resource regime. It comprises three components: contrastive training, monotonic chunk-wise attention and CNN-GRU-Softmax, where Monotonic Chunk-wise attention exploits the discrete and chunk level dependencies. We experimented on the datasets of four domains, Article, Conversation, Disease and Tourism, of the Hindi treebank, Tweet domain from TweeBank, Newswire domain from Penn TreeBank (PTB) and Tweet domain from ARK and compared it with several state-of-the-art models. We have obtained 96.63%, 94.34%, 91.24%, 93.76%, 92.30%, 97.51% and 93.55% accuracy on respective domains after CMCCGS has been applied. CMCCGS model has been further extended to domain adaptation by using single and multi-source domain adaptation to allow fine-tuning. It is analysed the effects on different layers. The extremely low resource domains such as Tourism, Disease and tweet domain of TweeBank and ARK have shown improvement in accuracy of +3.00%(96.76%) by an Article domain, +4.14%(95.38%) by Article and Tourism (multi-source), +2.93%(95.23%) by PTB domain and +1.43%(94.98%) by PTB and TweeBank (multi-source) as source domain, respectively. However, the Conversation domain has a negative impact on domain adaptation.

Keywords Part of Speech tagging · Domain adaptation · Contrastive training · Monotonic chunk-wise attention

✉ Rajesh Kumar Mundotiya
rajeshkm.rs.cse16@iitbhu.ac.in

Arpit Mehta
arpitmehta.cse18@iitbhu.ac.in

Rupjyoti Baruah
rupjyotibaruah.rs.cse18@iitbhu.ac.in

[1] Department of Computer Science and Engineering, IIT (BHU), Varanasi, India

1 Introduction

Part-of-speech (POS) tagging is one of the fundamental tasks in Natural Language Processing (NLP) and is a prerequisite for many practical applications as well as necessary for many tools in Computational Linguistics such as Chunking, Local Word Grouping and Parsing. POS tagging analyses the syntactic structure of the text and assigns grammatical categories to them.

Contextual semantics of words and phrases is necessary to disambiguate among the different possible categories of a token. Initial studies were based on standard rules (hand-crafted) that show sub-optimal performance in real-life practices. Until recently, the state-of-the-art (SOTA) was based on statistical learning algorithms, which are data-driven approaches. Like many other tasks, the SOTA is now based on deep neural networks. Being a data-driven approach, the optimal performance of such methods rely on the quantity of the annotated data, although they achieve high accuracies and robustness over the traditional SOTA methods. However, sufficient data for different domains may not be available for low-resource languages.

Since the majority of standard annotated data belongs to specific domains such as News, being trained on a specific domain, the trained model tested on other domains like Health, Social media, Literature, and Forum understandably gives a worse performance [7]. One major reason is that the source and target vocabularies are different and thus, some target words may never have appeared during the training phase. For instance, in the Hindi treebank dataset,[1] [2] disease-specific terminology such as disease names, symptom identifiers and treatments are frequent in the Health domain but are rare in the News and other domains. In order to generalise models to other domains, an optimal adaptation method is required that can transfer gained knowledge from the high resource domain to the low resource domain as much as possible.

In deep learning, the data is the only source of model training so that the expressivity of the language used depends solely on the domain of the data as far as the modelling process is concerned. That an equivalent performance will be obtained on other domains is not guaranteed due to the distinction between the distributions of domain-specific information [2] for different domains. While training the model, the model learns both general and domain-specific features [7]. These general or primitive features can be transferred to the target domain for either extracted features (through Transfer Learning [31]) or initialization (through Fine-tuning [40]). Fine-tuning is very effective as they use the available features from the pre-trained model through neurons [40] and allow their use for other domains to address problems like data scarcity.

However, due to fine-tuning, these pre-trained neurons are often biased towards the source data as the features are not entirely general or primitive and the distributions are still different. The initialization of general and domain-specific features adds to POS tagging's performance for other domains, where the source domain is considered a high resource and the other (target) domain is considered a low resource. Earlier methods have relied on source domain data, while recently proposed methods avail the target domain data as well through lexicons [41], monolingual corpora [20] and partial-labelled data [8, 34] to advancement into the performance of adaptation for sequence labelling. Such methods belong to semi-supervised or unsupervised adaptation.

[1] http://ltrc.iiit.ac.in/hutb_release/.

[2] https://ltrc.iiit.ac.in/showfile.php?filename=downloads/kolhi/.

When a learned model is employed from a source training data to a different but related target test data, it is pretended that both data accompany an equivalent distribution. When the distribution of both data is distinct, the DL model will not generalise well on the target domain. The model learning with the existence of an inequivalent distribution of data is known as Domain Adaptation [25]. The difference of inequivalent distribution is minimised by Contrastive learning. Contrastive learning is a self-supervised learning method that learns embedding space by contrasting semantically.

This paper focused on data-based, supervised domain adaptation, where we used a small amount of annotated datasets to perform single and multi-source domain adaptation by employing a novel deep learning-based architecture. This model is enhanced by contrastive training to make models for the low resource regimes more robust. The contributions of this paper are as follows:

1. We have proposed a novel deep learning-based architecture, Contrastive Monotonic Chunk-wise attention with CNN-GRU-Softmax (CMCCGS),[3] that leverages the features from existing state-of-the-art architecture and monotonic chunk-wise attention mechanism for low resource languages.
2. The proposed model architecture supports contrastive training, which yields better results on the target domain than the state-of-the-art architecture for this problem.
3. Based on the authors' knowledge, this is the first attempt at multi-source domain adaptation to POS tagging. The effect of single-source and multi-source domain adaptation on model performance has also been evaluated.
4. While performing domain adaptation, the effect of different layers on model performance has been evaluated.

The structure of this paper is as follows: Sect. 2 explains some related work on data and model-based domain adaptation. Section 3 describes the proposed model architecture with its components, such as contrastive training, monotonic chunk-wise attention and domain adaptation. Section 4 contains the description of datasets and experimental settings of the proposed and SOTA model training. The results obtained from these models are mentioned in Sect. 5.

2 Related Work

Initially, domain adaptation was successfully employed on certain problems such as language modelling, shallow parsing and capitalization. Chelba et al. [3] used the out of domain data considered as prior knowledge to estimate the probabilities of capitalization. They have used the maximum entropy-based techniques (probabilistic approaches (conditional and discriminative)) for capitalization as sequential labelling, where the prior distribution of out of domain data was estimated with the same model. Daumé et al. [9] designed the feature space in three ways. Source, target and general feature are first obtained, followed by blending the source and target data.

Miller et al. [28] used domain knowledge which includes derivational and inflectional morphology, and orthographic features for handling domain-specific words and unknown words. Further, the approach was extended to multiple-domain adaptation [27]. Some instances get higher weightage while performing domain adaptation training. Liu et al. [19] procured such instances from the target domain using a heuristic method of re-training the maximum

[3] https://github.com/Rajesh-Mundotiya/Journal-paper-adv-tl-monotonic-chunkwise-attn.

entropy model. Similarly, some standard and specific features of source and target domains were identified as pivot features by structural correspondence learning method [2].

Huang et al. [18], and Kruengkrai et al. [15] induce generalised features as latent features using a language model with a Hidden Markov Model (HMM), applied to multi-task learning such as POS tagging, word segmentation and Chunking. Liu et al. [21] proposed joint training, which is another variant of multi-task learning, which has performed well on segmentation and POS tagging by using self-training (semi-supervised) and clustering (unsupervised) methods. Zhang et al. [41] analyse the Chinese Treebank (CTB) model performance, trained with joint learning of word segmentation and POS tagging by tag dictionary, type, token and lexicon as features, and attained a slight improvement in accuracy over the only lexicon features for the target domain.

On the other hand, representation learning has been used as an alternative method to solve this problem, where the multi-dimensional vector represents words by using factorial HMM for probability estimation [16]. Ferraro et al. [11] have followed lexical probability rules in transformation-based learning algorithms for domain adaptation. The magnitude of the source and the target's probability distribution decides which techniques will apply to the task. Lexical bias is one of the ways to analyse the magnitudes, which can be addressed by supervised learning techniques [33]. Generalised features from representation learning and standard, domain-specific features from feature augmentation were obtained for enabling cross-domain model learning performed by Xiao et al. [38]. Schnabel et al. [34] proposed a robust domain adaptation method. This approach includes the local context (contextual word, word feature, suffix feature, shape feature) of the word as input to train the linear support vector machine.

The drastic effect on model performance with certain techniques is observed due to the co-variant shift (bias in marginal data distributions) so that the algorithm performs adequately in an adversarial learning environment. Globerson et al. [12] performed adversarial training with missing features where they assumed that missing features are distributed randomly in test data. Søgaard [35] targeted the most predictive features of the model by antagonistic adversaries. This method eliminated the essential features and did a frequent update of rare features (less confident). Guo et al. [14] trained the model by meta-learning in which a single source domain was treated as the meta-target and the remaining as the meta-source. The losses of these meta-learning expertise were minimised based on joint losses. This source and target domain encoding was aligned by adversarial training, where the target domain was treated as unlabelled. There is minimal work reported for POS tagging using adversarial training while used in various other applications associated with the domain adaptation [10, 36, 37].

3 Contrastive Monotonic Chunk-Wise Attention with CNN-GRU-Softmax (CMCCGS) Model Architecture

In this paper, a deep learning-based novel architecture is being proposed for domain adaptation for POS tagging. This architecture mainly consists of four components. The Convolutional Neural Network (CNN) is responsible for generating a word vector to preserve character-level information [17]. Bi-directional Gated Recurrent Unit (Bi-GRU) uses the available information to establish the dependencies among words at a time-step [5]. Monotonic CHunk-wise Attention (MOCHA) computes attention over the window size, and Softmax calculates the distribution of the label for a word [4]. The core model architecture,

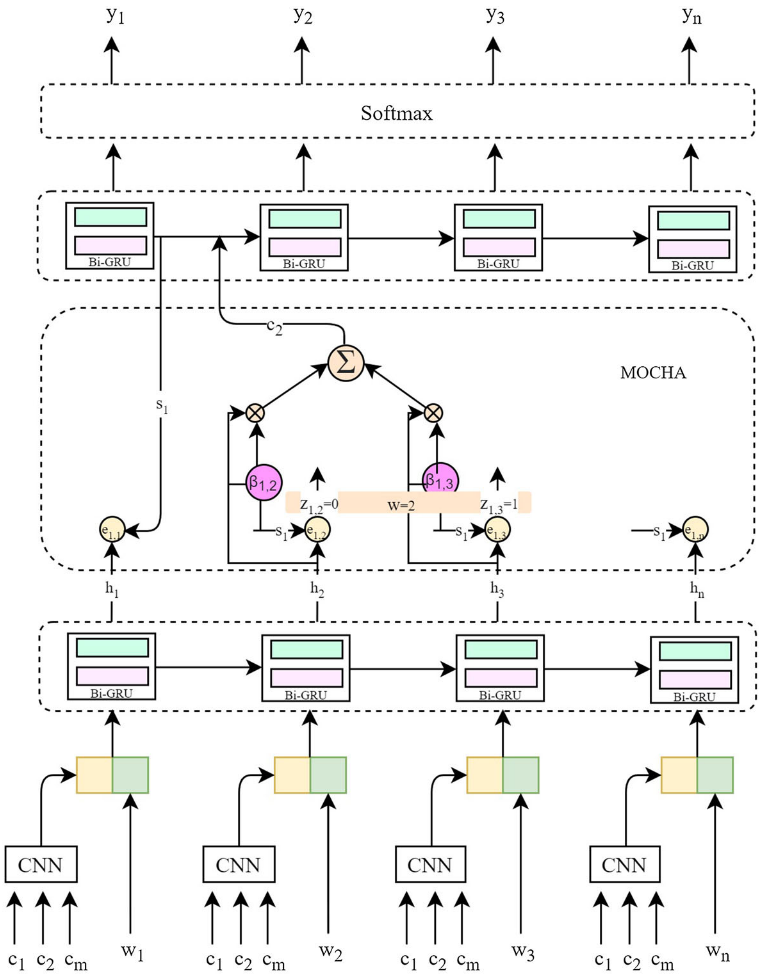

Fig. 1 The proposed MCCGS model architecture for POS tagging

called Monotonic Chunk-wise attention with CNN-GRU-Softmax (MCCGS), is illustrated in Fig. 1.

In this architecture, the tokens of the input sentences are used to generate the final word vector through a distributional word embedding and character-based word embedding by CNN. The Bi-GRU layer takes these as input to finding the relations among the words. These relation dependencies depend on contextual resemblance among the words. The MOCHA layer has been deployed over Bi-GRU's output on the assumption that phrases are constructed through contextual words. The obtained outputs from each attention have been passed to the Bi-GRU before calculating the final label distribution by Softmax. The core components of the model architecture, CNN, Bi-GRU and MOCHA, are described in Sects. 3.2 and 3.3. This model uses contrastive training (in Sect. 3.4) during learning. Hence, it is being called Contrastive Monotonic Chunk-wise attention with CNN-GRU-Softmax (CMCCGS). The way we perform domain adaptation by using this proposed model is explained in Sect. 3.5.

3.1 Problem Definitions

Let $\{S_p, Y_p\} \in X$ is a training instance, where p is a size of training space with $1 \leq p \leq N$. Here, s is a sentence and Y is a label. The s and Y themselves contain a sequence of words $s = \{w_1, w_2, \ldots, w_n\}$, and the corresponding set of labels are $Y = \{y_1, y_2, \ldots, y_n\}$. Each label in Y should belong to a predefined label set $\{l_1, l_2, \ldots, l_p\}$. The aim of POS tagging is to assign the labels $\{y_1, y_2, \ldots, y_n\}$ to the words $\{w_1, w_2, \ldots, w_n\}$ which appeared in a sentence s, which, in turn, is achieved through the objective function $f : S \rightarrow Y$.

3.2 CNN-Bi-GRU Component

Consider an input sentence as $S = \{w_1, w_2, \ldots, w_n\}$, where w_i is the word and n is the length of the sentence. A trainable fully connected layer which is known as the embedding layer, produces output as $\{x_1, x_2, \ldots, x_n\}$, where $x_i \in R^D$. The output from this layer is a sequence of vectors.

$$x_{1:n} = \mathbf{W} \cdot w_{1:n} \tag{1}$$

Another bottom layer of our proposed architecture (in Fig. 1) is that of character embedding, which aims to get a sequence of dense vectors by converting a word of a sentence w_i from a sequence of characters to a vector space model. The CNN component for the character embedding obtains the local context information by considering a filter, denoted by $F \in R^{KD}$, where K is the window size and D is character level one hot vector dimension. This filter learns the j^{th} character as the contextual representation by:

$$c_{i,j} = \phi\left(\mathbf{F} \cdot x_{i,\lfloor j-\frac{(K-1)}{2}\rfloor:\lfloor j+\frac{(K-1)}{2}\rfloor} + \mathbf{b}\right), \tag{2}$$

where ϕ is the non-linear activation function, which is $ReLU$, b is the learnable bias and $x_{\lfloor j-\frac{(K-1)}{2}\rfloor:\lfloor j+\frac{(K-1)}{2}\rfloor}$ denotes the concatenation of the embeddings of characters from $\lfloor j - \frac{(K-1)}{2}\rfloor$ to $\lfloor j + \frac{(K-1)}{2}\rfloor$ to i^{th} word.

Each input word w_i has a m number of characters. Multiple filters with different window sizes (varying from 2 to 5) are used to learn contextual character representations. The maximum pooling operation is applied over the convoluted features.

$$c_i = \{c_{i,j}\}_{j=1}^{m} = \max\{c_{i,1}, c_{i,2}, \ldots, c_{i,m-K+1}\} \tag{3}$$

After performing maximum pooling, the concatenation of the output of all filters is the contextual representation of the i^{th} word cw_i. The generated features cw_i have passed to three stacked fully connected layers. The resultant vector v_i from the penultimate layer has the character level information for the given i^{th} word.

$$v_i = \text{ReLU}(\mathbf{W} \cdot cw_i + \mathbf{b}) \tag{4}$$

The $\mathbf{W}$ and $\mathbf{b}$ are learning parameters. The $v = \{v_1, v_2, \ldots, v_n\}$ is the output of the CNN layer. These character vectors v_i are stacked with the word embedding vector x_i to generate a final representation of the word vector cx_i.

$$cx_i = [x_i; v_i] \tag{5}$$

The intra-word dependencies have been calculated by the temporal sequence method. One of them is the Gated Recurrent Unit (GRU). GRU overcomes the gradient vanishing problem to a certain extent. However, we have used Bidirectional GRU, which provides more robust learning for longer dependencies. The input sentence has been represented into vector space CX, where $CX = \{cx_1, cx_2, \ldots, cx_n\}$. These input sentence vectors are converted into hidden states $h = \{h_1, h_2, \ldots, h_n\}$ by applying Bi-GRU:

$$h_{1:n} = \overrightarrow{\text{GRU}}(CX_{1:n}) \oplus \overleftarrow{\text{GRU}}(CX_{1:n}) \tag{6}$$

3.3 MOCHA Component

MOCHA [4] can estimate attention over conditional preceding memory units, unlike soft attention, which considers all the memory units. The term monotonicity refers to the computation of the attention score based on the hidden states of the timesteps. Those timesteps that are a subset of the whole input state sequence generated according to the input window length are known as chunks.

In tag's $j - 1$ timestamp, the hidden state is represented by s_{j-1} for compiling the unnormalized energy function $e_{j,i}$, defined as:

$$e_{j,i} = \text{MonotonicEnergy}(s_{j-1}, h_i) \tag{7}$$

where

$$\text{MonotonicEnergy}(s_{j-1}, h_i) = \mathbf{g}\frac{\mathbf{v}^T}{||\mathbf{v}||} \tanh\left(\mathbf{W} \cdot s_{j-1} + \mathbf{W} \cdot h_i + \mathbf{b}\right) + \mathbf{r} \tag{8}$$

and $\mathbf{g}$, $\mathbf{v}$, $\mathbf{W}$, $\mathbf{W}$, $\mathbf{b}$ and $\mathbf{r}$ are the parameters which are learnt during training. The selection probability p_{ji} is computed using the logistic sigmoid function.

$$p_{ji} = \sigma(e_{ji} + \epsilon); \epsilon \in \ N(0, 1) \tag{9}$$

Here ϵ, the unit variance Gaussian distribution with zero mean, is considered noise to produce robust binary values. The state selection for the window depends on the output of $z_{j,i} = \text{Bernoulli}(p_{ji})$, which is further used for context vector generation. The monotonic attention probability has been calculated by using the selection probability:

$$\alpha_{j,:} = p_{j,:} \cdot \text{cumprod}(1 - p_{j,:}) \cdot \text{cumsum}\left(\frac{\alpha_{j-1,:}}{\text{cumprod}(1 - p_{j,:})}\right) \tag{10}$$

This ChunkEnergy is used to calculate the attention score. The score is calculated by:

$$u_{j,i} = \text{ChunkEnergy}(s_{j-1}, h_i) \tag{11}$$

$$\text{ChunkEnergy}(s_{j-1}, h_i) = \mathbf{v}^T \cdot \tanh(\mathbf{W} \cdot h_i + \mathbf{W} \cdot s_{j-1} + \mathbf{b}) \tag{12}$$

where $\mathbf{v}, \mathbf{b}$ and $\mathbf{W}$ are the learning parameters. The Softmax function normalizes the ChunkEnergy over the selected chunk of w, which is calculated by MovingSum. The probability distribution of MOCHA $\beta_{j,:}$ is computed as:

$$\beta_{j,:} = \exp(u_{j,:}) \cdot \text{MovingSum}\left(\frac{\alpha_{j,:}}{\text{MovingSum}(\exp(u_{j,:}), w, 1)}, 1, w\right) \tag{13}$$

$$\text{MovingSum}(x, b, f)_n = \sum_{m=n-b+1}^{n+f-1} x_m \tag{14}$$

Finally, the context vector's expected value for j^{th} timestamp is calculated by normalizing to each hidden state of the input sentence vector:

$$c_j = \sum_{i=1}^{n} \beta_{j,i} \cdot h_i \tag{15}$$

This context vector is merged with the previous hidden state of the previous timestamp fed to Bi-GRU. The hidden representation of this is used to calculate the probability distribution of the labels by using Softmax:

$$s_{i,:} = \overrightarrow{\mathrm{GRU}}(c_{j,:}, s_{i-1,:}) \oplus \overleftarrow{\mathrm{GRU}}(c_{j,:}, s_{i-1,:}) \tag{16}$$

$$p(Y|S;\boldsymbol{\theta}) = p(y_1, y_2, \ldots, y_n | w_1, w_2, \ldots, w_n; \boldsymbol{\theta}) = \frac{exp(\mathbf{W} \cdot s_{i,:} + \mathbf{b})}{\sum_{l=1}^{n} exp(\mathbf{W} \cdot s_l + \mathbf{b})} \tag{17}$$

Where $\boldsymbol{\theta}$ represents all of the model parameters (in BiGRUs, Softmax, Dense, and MOCHA). These layers implicitly perform feature extraction and classification tasks during the model training with the parameters of θ_f and ϕ_c, respectively. The summation of both parameters, i.e., θ is minimised during the training of the model by the negative log-likelihood loss function:

$$L(\boldsymbol{\theta}; S, Y) = -\log \cdot p(Y|S; \boldsymbol{\theta}) \tag{18}$$

3.4 Contrastive Training Component

One of the earlier attempts of using contrastive training in NLP applications (text classification) is reported by Miyato et al. (2017) [29]. They encoded perturbation into the word embedding to generate a contrastive example. More robustness in the classifier was obtained after adding perturbations (η) to the input examples, which are closer to actual examples. These perturbed examples have the same label as the actual example to which they are close. However, it may be misclassified by the current classifier, which increases the loss, so the current model loss is modified by:

$$\eta = \underset{\eta' : \|\eta'\|2 \le \delta}{\arg\max} \; L(\hat{\theta}; S + \eta', Y) \tag{19}$$

Here, the normalised perturbation δ allows to construct the contrastive example for each training step:

$$S_{adv} = S + \eta \tag{20}$$

The new loss function ($\tilde{L_\theta}$) of the contrastive part is based on classifier having a mixture of actual examples $L(\theta; S, Y)$ and perturbed examples' loss $L(\theta; S_{adv}, Y)$, where individual loss is controlled by the weighting factor γ with 0.5.

$$\tilde{L_\theta} = \gamma L(\theta; S, Y) + (1 - \gamma) L(\theta; S_{adv}, Y) \tag{21}$$

The pseudo-code of the CMCCGS learning algorithm is described in algorithm 1.

3.5 Domain Adaptation

Domain adaptation approaches follow the data or model-oriented techniques with unsupervised, semi-supervised and supervised settings for diverse applications of NLP, including POS tagging. Here, we have exploited the fine-tuning approach for performing domain adaptation, where the learned features from the source domain are transferred at the time of the

Algorithm 1 Contrastive Monotonic Chunk-wise attention with CNN-GRU-Softmax (CMC-CGS) for POS tagging

```
Require: $N$: number of training examples, i.e., sentences, $w$: chunk size, $\epsilon$: emission score $\theta$ ←Random initialize
while s = 1,2,...,N do
    Obtain word vectors $x_{i:n}$ by Eq. (1)
    for $i = 1$ to $n$ do
        Obtain character level word vector $v_i$ by Eq. (4)
    end for
    Obtain word embedding by concatenation according to Eq. (5)
    Obtain hidden states $h_{1:m}$ on word embedding by Eq. (6)
    for $i = 1$ to $n$ do
        for $j = 1$ to $n$ do
            Calculate chunk energy $u_{j,i}$ with $w$ and $\epsilon$ by Eq. (11)
        end for
        Obtain contextual attention vector $c_j$ by Eq. (15)
        Calculate the hidden vector and probability score to each label by Eq. (16) and Eq. (17)
    end for
    Generate contrastive examples by Eq. (20)
    Calculate loss considering $\theta$ and label predictions according to Eq. (21)
    $i \leftarrow i + 1$
end while
```

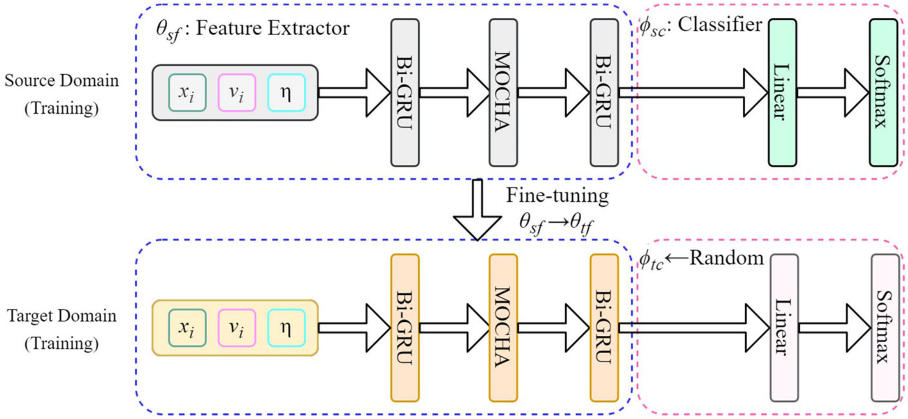

Fig. 2 Domain adaptation

target domain learning. This feature transfer procedure closely follows the Meftah et al. [26] settings. In this procedure, the entire model has been trained on the source domain for POS tagging and learns the optimal learned parameters.

Each domain dataset has different labels, so the training parameters of the proposed model have been interpreted as label-aware parameters (ϕ_c) and data-aware parameters (θ_f), which attain classification and feature extraction, respectively. Feature extractor parameters are considered transferable due to their generality, which considers MOCHA and both Bi-GRU layers. Linear and Softmax layers are considered classification parameters that are task specific. Based on the training parameters, the model has split into two parts: Feature Extractor and Classifier, as shown in the domain adaptation settings in Fig. 2.

The learnt parameters of Feature Extractor, θ_{sf} from source domain are used as parameter initialization to Feature Extractor of the target domain, θ_{tf} during domain adaptation. The

Algorithm 2 Domain adaptation for POS tagging

```
Require: Target domain data, $\theta_{sf}$ : Learnt parameters (source domain) from Algorithm 1 for Feature Extractor
$\phi_{tc}$ ← Randomly initialized parameters (target domain) for Classifier
while available target domain data do
    $\theta_{tf} \leftarrow \theta_{sf}$
    for each domain data do
        $\theta_{s'f}$ ← Train Algorithm 1 with $\theta_{tf}$ and $\phi_{tc}$
    end for
    $\theta_{sf} \leftarrow \theta_{s'f}$
end while
```

Table 1 Dataset statistics for each domain

Dataset	Domain	Training Data	Testing Data	Total
HT	Article	15088	1910	16998
HT	Tourism	2400	622	3022
HT	Conversation	1700	496	2196
HT	Disease	900	594	1494
PTB	Newswire	55000	12499	67499
TweeBank	Tweet	2800	750	3550
ARK	Tweet	1700	675	2375

remaining parameters of Classifier, ϕ_{tc}, is randomly initialized. After initialization ($\theta_{sf} \rightarrow \theta_{tf}$ and $Random \rightarrow \phi_{tc}$), the model starts learning from previously learned features of the source domain, named as single source domain adaptation, which is shown in Fig. 2. During the training of domain adaptation, the Classifier parameters have trained on the target domain dataset from scratch. The same procedure follows with multi-source settings, where the Feature Extractor parameters have been transferred from first source to second source ($\theta_{sf} \rightarrow \theta_{s'f}$) and second source to target source ($\theta_{s'f} \rightarrow \theta_{tf}$). The pseudo-code of the domain adaptation is described in algorithm 2.

4 Experimental Descriptions

4.1 Dataset Descriptions

For most languages, annotation data is not available due to low resource constraints. Moreover, the availability of a domain-specific annotated dataset in these languages is rare. Here, we have exploited Hindi Treebank (HT) [1], Penn-TreeBank (PTB) [24] of Wall street journal (WSJ), ARK [30] and the recent TweeBank [22] dataset. The HT dataset includes Article, Conversation, Disease and Tourism domain with 32, 32, 39 and 46 POS tags. The PTB, ARK and TweeBank include Newswire and Tweet domains with 45, 26 and 18 POS tags. The comparative statistics of training and testing for each dataset with each domain are depicted in Table 1. We evaluated the state-of-the-art models and CMCCGS model on each domain of the datasets.

4.2 Experimental Settings for Models

Initially, we evaluated the state-of-the-art models and CMCCGS model on each domain of the datasets. The maximum length of words and sentences has been fixed for training the model, which is 22 and 52, respectively. However, gradient calculation avoided the padded sentences and words, which in turn prevents overfitting. The character vector size 32 is obtained after applying two filters, 64 and 124, each with the size of 3, with a dropout of 30%. The model trained with the word vector and GRU unit of 100 and 128, respectively. The chunk size is 10. As the annotation corpus is tiny, the model tends to overfit quickly. Hence, dropout and early stoppage have been applied with the value of 50% and 5 as patience, respectively. The model has been trained using Adam optimiser with an initial learning rate of 0.008, which further decays over each epoch by 0.005 for Disease and Tourism. It is tuned to 0.01 as the learning rate and 0.007 as the decay rate for Article, Conversation, PTB, ARK and TweeBank. The SOTA models have trained with their default settings.

5 Results and Analysis

The standard evaluation metrics such as Precision (Pre), Recall (Re), F1-score (F1) and Accuracy (Acc) have been considered to evaluate the proposed model. Tables 2 and 3 shows the results obtained from different domains of HT dataset and the PTB, ARK and TweeBank dataset, respectively. These results are compared with state-of-the-art models, which are neural network-based sentence-level log-likelihood (NN-SLL) [6] and Bi-directional LSTM-CNNs-CRF (BLSTM-CNN-CRF) [23].

As the size of the Article is maximal, the CMCCGS model (96.63%) provided a comparable accuracy score to state-of-the-art models, which are 94.11% and 96.64%. The highest score is 91.24% and 94.34% obtained for Disease and Conversation using the CMCCGS model, respectively, since the size of the dataset of these domains is minimal. In contrast to the NN-SLL model, the CMCCGS model has reported a lower score, 93.76%, for the Tourism domain. It can be seen that the CMCCGS approach performed better on the Conversation domain. The difference between state-of-the-art (SOTA) models and CMCCGS for Article is very slight, but a significant improvement has been observed for the rest of the domains, as shown in Table 2. On the other hand, Table 3 showed that PTB and ARK obtained the highest accuracy score, which are 97.51% and 93.55% through the CMCCGS model, while the BLSTM-CNN-CRF yields 93.05% for TweeBank. It can be clearly interpreted through Fig. 3. Contrastive training surpasses the performance of current top-performing models [23, 32] for POS tagging in different domains [13, 39]. Hence, the performance is gained on the small size domains after applying the contrastive and adversarial based models.

5.1 Domain Adaptation Results

It can be understood from Tables 2 and 3 that CMCCGS provides robust performance on the minimal datasets. Therefore, the same (hyper-)parameter settings have been employed to perform domain adaptation. We have assumed that the source domain always has larger training data than the target domain. For example, Disease as the target domain considered for Article, Tourism and Conversation domain, Conversation as target domain considered for Article and Tourism, and Tourism as target domain considered for Article domain of HT

Table 2 Results obtained on each domain of HT dataset

Domain	Model	Pre	Re	F1	Acc
Article	NN-SLL [6]	93.93	94.11	93.89	94.11
	BLSTM-CNN-CRF [23]	96.57	96.65	96.57	96.64
	CMCCGS	96.09	96.62	96.35	96.63
Tourism	**NN-SLL** [6]	95.13	95.03	94.93	95.03
	BLSTM-CNN-CRF [23]	93.42	93.37	93.09	93.37
	CMCCGS	93.76	93.51	93.69	93.76
Conversation	NN-SLL [6]	87.87	90.37	88.84	90.37
	BLSTM-CNN-CRF [23]	91.59	91.65	91.43	91.64
	CMCCGS	94.03	94.35	94.01	94.34
Disease	NN-SLL [6]	91.14	91.24	90.86	91.24
	BLSTM-CNN-CRF [23]	90.74	90.76	90.34	90.89
	CMCCGS	91.91	91.25	91.30	91.24

Table 3 Results obtained on PTB, TweeBank and ARK datasets

Data	Model	Pre	Re	F1	Acc
PTB	NN-SLL [6]	96.55	96.58	96.53	96.58
	BLSTM-CNN-CRF [23]	97.38	97.36	97.37	97.36
	CMCCGS	97.42	97.51	97.41	97.51
TweeBank	NN-SLL [6]	93.34	92.87	92.92	92.87
	BLSTM-CNN-CRF [23]	93.70	93.70	93.05	93.05
	CMCCGS	93.30	92.30	92.18	92.30
ARK	NN-SLL [6]	93.96	92.67	92.38	92.67
	BLSTM-CNN-CRF [23]	93.48	93.32	93.28	93.32
	CMCCGS	94.32	93.55	93.69	93.55

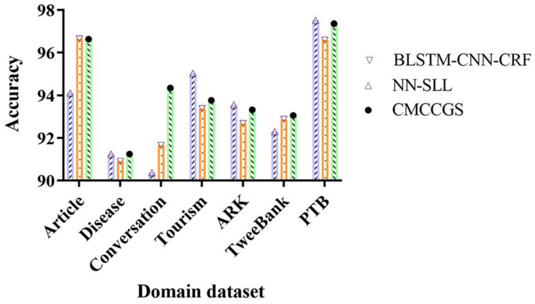

Fig. 3 Accuracy comparison of SOTA and CMCCGS models

dataset. The PTB is considered the source domain for ARK and TweeBank. The TweeBank has smaller training data compared to PTB. Thus it is also the target domain for PTB.

We have compared domain adaptation results with the SOTA model, Domain adaptation using Hierarchical Bidirectional LSTM-CRF (HBLSTMC)[4] [25]. After domain adaptation,

[4] This abbreviation is used to compare our results.

Table 4 Domain adaptation on the HT dataset

Source	Target	Model	Pre	Re	F1	Acc
Article	Conversation	HBLSTMC [25]	90.66	92.35	91.16	92.35
		CMCCGS	92.79	93.41	92.73	93.40
Tourism	Conversation	**HBLSTMC** [25]	92.51	92.92	92.39	92.92
		CMCCGS	91.79	92.76	91.77	92.75
Article	Disease	HBLSTMC [25]	90.97	91.86	90.50	91.86
		CMCCGS	93.90	94.22	93.87	94.22
Tourism	Disease	**HBLSTMC** [25]	95.75	95.88	95.75	95.88
		CMCCGS	92.41	92.56	92.28	92.55
Conversation	Disease	**HBLSTMC** [25]	94.91	94.53	94.52	94.53
		CMCCGS	91.87	91.82	91.60	91.82
Article	Tourism	HBLSTMC [25]	95.16	95.74	95.26	95.74
		CMCCGS	96.64	96.76	96.62	96.76

Table 5 Domain adaptation on ARK and TweeBank datasets

Source	Target	Model	Pre	Re	F1	Acc
PTB	ARK	HBLSTMC [25]	95.00	94.36	94.67	94.36
		CMCCGS	94.57	94.70	94.54	94.70
PTB	TweeBank	**HBLSTMC** [25]	96.22	96.23	96.15	96.23
		CMCCGS	95.11	95.23	95.03	95.23

the maximum score has been obtained for Conversation (93.40%) and Tourism (96.76%) by CMCCGS and Disease (95.88%) by HBLSTMC. Similarly, Table 5 showed that CMCCGS works adequately for ARK while HBLSTMC model for TweeBank.

The significance of training data to our proposed supervised domain adaptation, the CMCCGS model, is observed on the HT dataset in Table 4. The Conversation and Disease domain attained the highest score when Article considered as source domain. The comparison of Table 2 with Table 4 gives the significance of domain adaptation where the effect on the performance of Tourism, Conversation and Disease are +3.00%, −0.94% and +2.98%, respectively. On the other hand, the performance has increased by +2.93% for TweeBank and +1.15% for ARK, as is obtained when comparing Table 3 with Table 5.

5.1.1 Multi-source Domain Adaptation

The deep learning model requires abundant data to provide promising results. The target domain has minimal data in our experiments. To address this, we have used the data from different domains to perform domain adaptation. The Disease domain has limited data compared to the others in HT dataset and it is considered the target domain and the remaining domains as the source domains for performing multi-source domain adaptation. Similarly, the Tweet domain from ARK is considered the target domain and the Newswire domain from PTB and Tweet domain from TweeBank are considered source domains. Based on the annotation statistics, these datasets can be divided into three categories, namely High (Arti-

Table 6 Results of Multi-source domain adaptation by CMCCGS model

Source (High+Moderate)	Target	Pre	Re	F1	Acc
Article+Conversation	Disease	92.34	92.82	92.29	92.81
Article+Tourism	**Disease**	95.21	95.39	95.11	95.38
Tourism+Conversation	Disease	93.35	93.58	93.30	93.58
Article+Tourism+Conversation	Disease	93.39	93.01	93.19	93.01
PTB+TweeBank	ARK	94.29	94.98	94.63	94.98

cle, PTB), Moderate (Tourism, Conversation and TweeBank) and Low (Disease and ARK) resources. According to these categories, the following two settings are used:

1. Model training initialised with High category (Article and PTB) and then the learned features were transferred to the Moderate category (Tourism or Conversation and TweeBank), which further applied domain adaptation to the target domain.
2. Model training initialised with High category (Article) and then the learned features were transferred to the Moderate category (Tourism). These learned parameters were then transferred to another Moderate category (Conversation) to apply domain adaptation on the target domain.

From these two experiments, we obtained the best result on the Disease and ARK by using Article with Tourism and PTB with TweeBank as source settings, compared to the other, as shown in Table 6.

5.2 Layers Freezing Results in Domain Adaptation

The proposed model, CMCCGS, has multiple core layers, i.e., Bi-GRU, MOCHA, and Softmax, which are responsible for capturing different input information. Hence, while performing the domain adaptation via the data-based Transfer learning approach, gradual layers have been fixed to show the model performance. The model layers have been divided into two sub-parts: Layers between the input layer to MOCHA layer, which includes the first Bi-GRU layer, referred to as part(i), and Layers after the MOCHA layer to the inference layer, i.e., the second Bi-GRU, the Dense, and the Softmax layer, referred as part(ii).

The effect on evaluation matrices of both sub-parts being frozen to each domain is shown in Table 7. The part(i) layers freezing of domain adaptation for Tourism from Article, for Conversation from Tourism, for ARK from PTB and TweeBank (multi-source) achieved better accuracy than part(ii) layers freezing. The part(ii) layer freezing benefited for Disease from multi-source domain adaptation (Article and Tourism) and TweeBank from PTB. However, the freezing scores of both sub-parts do not yield better results than the full model training while performing domain adaptation, as shown in Fig. 4.

The training of the two sub-parts being frozen shows up as the differences in the metrics scores for each domain on each kind of domain adaptation as follows:

1. The differences for the Disease domain lie in the ranges of −0.37 to +1.85, −0.23 to +1.74, −0.20 to +1.77 and −0.23 to +1.74 for precision, recall, F1-score and accuracy, respectively.
2. The differences for the Conversation domain are in the ranges of −1.52 to +0.04, −0.71 to +0.02, −1.18 to +0.14 and −0.71 to +0.01 for the respective metrics.

Table 7 Results of the domain adaptation based on freezing layers' by CMCCGS model

Source	Target	Freeze	Pre	Re	F1	Acc
Article	Disease	part(i)	92.54	92.96	92.76	92.96
Article	Disease	part(ii)	92.42	92.71	92.24	92.71
Article	**Tourism**	**part(i)**	96.40	96.30	96.09	96.30
Article	Tourism	part(ii)	95.76	96.18	95.97	96.18
Article	Conversation	part(i)	90.26	92.15	90.69	92.14
Article	Conversation	part(ii)	91.78	92.25	91.87	92.24
Tourism	**Conversation**	**part(i)**	91.66	92.87	91.96	92.86
Tourism	Conversation	part(ii)	91.62	92.85	91.82	92.85
Tourism	Disease	part(i)	94.38	94.51	94.21	94.50
Tourism	Disease	part(ii)	93.48	93.65	93.44	93.64
Conversation	Disease	part(i)	89.64	89.45	89.44	89.45
Conversation	Disease	part(ii)	89.18	88.92	88.81	88.92
Article+Tourism	Disease	part(i)	94.43	94.74	94.45	94.73
Article+Tourism	**Disease**	**part(ii)**	94.80	94.97	94.66	94.96
Article+Conversation	Disease	part(i)	91.45	92.07	91.50	92.07
Article+Conversation	Disease	part(ii)	91.09	91.49	91.04	91.48
Tourism+Conversation	Disease	part(i)	92.82	93.07	92.77	93.07
Tourism+Conversation	Disease	part(ii)	90.97	91.33	91.00	91.33
Article+Tourism+Conversation	Disease	part(i)	94.46	94.68	94.36	94.67
Article+Tourism+Conversation	Disease	part(ii)	94.52	94.74	94.42	94.74
PTB	ARK	part(i)	94.65	94.50	94.34	94.50
PTB	ARK	part(ii)	92.37	93.01	92.39	93.01
PTB	TweeBank	part(i)	92.82	92.17	92.07	92.17
PTB	**TweeBank**	**part(ii)**	93.98	93.48	93.37	93.48
PTB+TweeBank	**ARK**	**part(i)**	94.54	94.51	94.32	94.51
PTB+TweeBank	ARK	part(ii)	93.30	93.59	93.14	93.59

3. Similarly, the Tourism domain has single domain adaptation experiments; hence it yields a difference of +0.64 for precision and +0.12 for all the remaining metrics.

Where negative digits indicate, part(ii) freezing yields a better result than part(i) freezing and vice-versa for positive digits.

6 Conclusions

Part of Speech (POS) tagging is a preliminary task of Natural Language Processing, which assigns the grammatical category to a word. It is still a challenging task for low resource languages. This paper presented a Fine-tuning based domain adaptation model for POS tagging on the available minimal annotated data of four domains, Article, Tourism, Conversation and Disease of Hindi treebank and Newswire and Tweet domain of PTB and ARK, TweeBank, respectively. The presented model consists of monotonic chunk-wise attention, bidirectional GRU and Softmax, named as Contrastive Monotonic Chunk-wise attention with CNN-GRU-

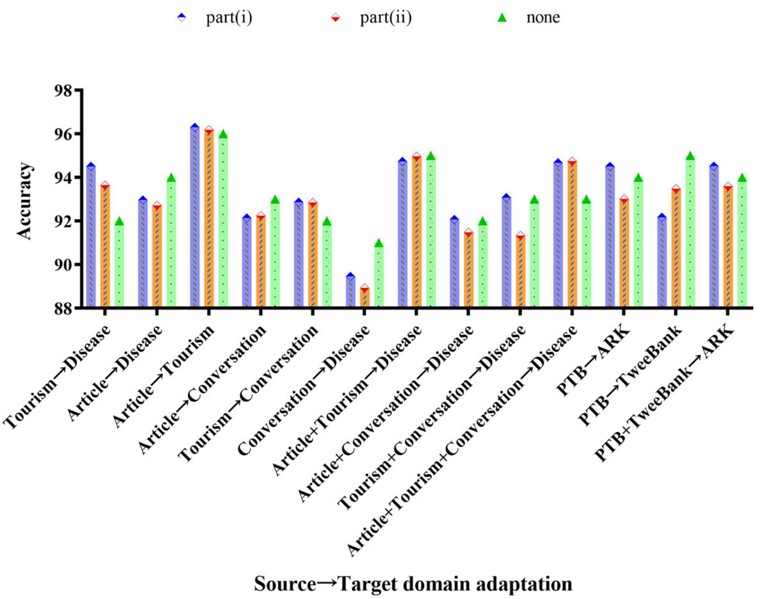

Fig. 4 Accuracy comparison of the CMCCGS model with (part(i) and part(ii)) and without (none) freezing layers'

Softmax (CMCCGS). This model is trained by contrastive training and performs single source and multi-source domain adaptation. The proposed model achieved state-of-the-art performance on minimal annotated training data. The obtained best accuracies' are 94.34% on the Conversation domain by contrastive training, and 96.76% on the Tourism domain by single-source domain adaptation, where Article is the source domain, 95.38% on a Disease domain by using multi-source (Tourism and Article both are considered as source), 94.98% on ARK by using PTB and TweeBank as source and 95.23% on TweeBank by using PTB as source domain adaptation. The tagset used in annotations of these domain datasets are the same, thus allowing domain adaptation. However, the model considers some common tags as different tags during domain adaptation, which suppresses the model performance for high resource domains, i.e., Article and PTB. This problem can be overcome by a mapping function that could be a tunable mapping function such as adversarial-discriminator and multi-task or a hard mapping function by exerting linguistic information, considered as future work.

References

1. Bhat RA, Bhatt R, Farudi A, Klassen P, Narasimhan B, Palmer M, Rambow O, Sharma DM, Vaidya A, Vishnu SR et al (2017) The hindi/urdu treebank project. In: Handbook of Linguistic Annotation. Springer, pp 659–697
2. Blitzer J, McDonald R, Pereira F (2006) Domain adaptation with structural correspondence learning. In: Proceedings of the 2006 conference on empirical methods in natural language processing, pp 120–128
3. Chelba C, Acero A (2006) Adaptation of maximum entropy capitalizer: little data can help a lot. Comput Speech Lang 20(4):382–399
4. Chiu C, Raffel C (2018) Monotonic chunkwise attention. In: 6th international conference on learning representations, ICLR 2018, Vancouver, BC, Canada, April 30–May 3, 2018, Conference track proceedings. OpenReview.net . https://openreview.net/forum?id=Hko85plCW

5. Cho K, van Merrienboer B, Gülçehre Ç, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Moschitti A, Pang B, Daelemans W (eds) Proceedings of the 2014 conference on empirical methods in natural language processing, EMNLP 2014, October 25–29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pp 1724–1734. ACL . https://doi.org/10.3115/v1/d14-1179
6. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P (2011) Natural language processing (almost) from scratch. J Mach Learn Res 12:2493–2537
7. Daumé III H (2007) Frustratingly easy domain adaptation. In: Proceedings of the 45th annual meeting of the association of computational linguistics. Association for computational linguistics, Prague, Czech Republic, pp 256–263. https://www.aclweb.org/anthology/P07-1033
8. Daumé III H, Kumar A, Saha A (2010) Frustratingly easy semi-supervised domain adaptation. In: Proceedings of the 2010 workshop on domain adaptation for natural language processing. Association for Computational Linguistics, pp 53–59
9. Daumé III H, Marcu D (2005) Learning as search optimization: approximate large margin methods for structured prediction. In: Proceedings of the 22nd international conference on machine learning, pp 169–176
10. Du C, Sun H, Wang J, Qi Q, Liao J (2020) Adversarial and domain-aware bert for cross-domain sentiment analysis. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 4019–4028
11. Ferraro JP, Daumé H III, DuVall SL, Chapman WW, Harkema H, Haug PJ (2013) Improving performance of natural language processing part-of-speech tagging on clinical narratives through domain adaptation. J Am Med Inf Assoc 20(5):931–939
12. Globerson A, Roweis S (2006) Nightmare at test time: robust learning by feature deletion. In: Proceedings of the 23rd international conference on Machine learning, pp 353–360
13. Gui T, Zhang Q, Huang H, Peng M, Huang X (2017) Part-of-speech tagging for Twitter with adversarial neural networks. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Copenhagen, Denmark, pp 2411–2420. https://doi.org/10.18653/v1/D17-1256.https://www.aclweb.org/anthology/D17-1256
14. Guo J, Shah DJ, Barzilay R (2018) Multi-source domain adaptation with mixture of experts. arXiv preprint arXiv:1809.02256
15. Huang F, Yates A (2009) Distributional representations for handling sparsity in supervised sequence-labeling. In: Proceedings of the joint conference of the 47th annual meeting of the ACL and the 4th international joint conference on natural language processing of the AFNLP. Association for Computational Linguistics, vol 1, pp 495–503
16. Huang F, Yates A (2010) Exploring representation-learning approaches to domain adaptation. In: Proceedings of the 2010 workshop on domain adaptation for natural language processing. Association for Computational Linguistics, pp 23–30
17. Kim Y (2014) Convolutional neural networks for sentence classification. In: A. Moschitti, B. Pang, W. Daelemans (eds.) Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL. ACL, pp 1746–1751. https://doi.org/10.3115/v1/d14-1181
18. Kruengkrai C, Uchimoto K, Kazama J, Wang Y, Torisawa K, Isahara H (2009) An error-driven word-character hybrid model for joint chinese word segmentation and pos tagging. In: Proceedings of the joint conference of the 47th annual meeting of the ACL and the 4th international joint conference on natural language processing of the AFNLP. Association for Computational Linguistics, vol 1, pp 513–521
19. Liu K, Chapman W, Hwa R, Crowley RS (2007) Heuristic sample selection to minimize reference standard training set for a part-of-speech tagger. J Am Med Inf Assoc 14(5):641–650
20. Liu Y, Zhang Y (2012) Unsupervised domain adaptation for joint segmentation and POS-tagging. In: Proceedings of COLING 2012: Posters. The COLING 2012 Organizing Committee, Mumbai, India, pp 745–754. https://www.aclweb.org/anthology/C12-2073
21. Liu Y, Zhang Y (2012) Unsupervised domain adaptation for joint segmentation and pos-tagging. In: Proceedings of COLING 2012: Posters, pp 745–754
22. Liu Y, Zhu Y, Che W, Qin B, Schneider N, Smith NA (2018) Parsing tweets into universal dependencies. In: Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, vol 1 (Long Papers), pp 965–975
23. Ma X, Hovy E (2016) End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In: Proceedings of the 54th annual meeting of the association for computational linguistics. Association for Computational Linguistics, Berlin, Germany, (vol 1: Long Papers), pp 1064–1074. https://doi.org/10.18653/v1/P16-1101. https://www.aclweb.org/anthology/P16-1101

24. Marcus MP, Marcinkiewicz MA, Santorini B (1993) Building a large annotated corpus of english: the penn treebank. Comput Linguist 19(2):313–330
25. März L, Trautmann D, Roth B (2019) Domain adaptation for part-of-speech tagging of noisy user-generated text. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, vol 1 (Long and Short Papers), pp 3415–3420
26. Meftah S, Semmar N (2018) A neural network model for part-of-speech tagging of social media texts. In: Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)
27. Miller J, Torii M, Vijay-Shanker K (2007) Adaptation of pos tagging for multiple biomedical domains. In: Biological, translational, and clinical language processing, pp 179–180
28. Miller JE, Bloodgood M, Torii M, Vijay-Shanker K (2006) Rapid adaptation of pos tagging for domain specific uses. In: Proceedings of the HLT-NAACL BioNLP workshop on linking natural language and biology. Association for Computational Linguistics, pp 118–119
29. Miyato T, Dai AM, Goodfellow I (2017) Adversarial training methods for semi-supervised text classification. In: ICLR
30. Owoputi O, O'Connor B, Dyer C, Gimpel K, Schneider N, Smith NA (2013) Improved part-of-speech tagging for online conversational text with word clusters. In: Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies, pp 380–390
31. Pan SJ, Yang Q (2009) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359
32. Plank B, Søgaard A, Goldberg Y (2016) Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Berlin, Germany (vol 2: Short Papers), pp 412–418. https://doi.org/10.18653/v1/P16-2067. https://www.aclweb.org/anthology/P16-2067
33. Schnabel T, Schütze H (2013) Towards robust cross-domain domain adaptation for part-of-speech tagging. In: Proceedings of the sixth international joint conference on natural language processing. Asian Federation of Natural Language Processing, Nagoya, Japan, pp 198–206. https://www.aclweb.org/anthology/I13-1023
34. Schnabel T, Schütze H (2014) Flors: fast and simple domain adaptation for part-of-speech tagging. Trans Assoc Comput Linguist 2:15–26
35. Søgaard A (2013) Part-of-speech tagging with antagonistic adversaries. In: Proceedings of the 51st annual meeting of the association for computational linguistics (vol 2: Short Papers), pp 640–644
36. Vu TT, Phung D, Haffari G (2020) Effective unsupervised domain adaptation with adversarially trained language models. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, Online, pp 6163–6173. https://doi.org/10.18653/v1/2020.emnlp-main.497. https://www.aclweb.org/anthology/2020.emnlp-main.497
37. Wright D, Augenstein I (2020) Transformer based multi-source domain adaptation. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, pp 7963–7974. https://doi.org/10.18653/v1/2020.emnlp-main.639. https://www.aclweb.org/anthology/2020.emnlp-main.639
38. Xiao M, Guo Y (2013) Domain adaptation for sequence labeling tasks with a probabilistic language adaptation model. In: International conference on machine learning, pp 293–301
39. Yasunaga M, Kasai J, Radev D (2018) Robust multilingual part-of-speech tagging via adversarial training. In: Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies. Association for Computational Linguistics, New Orleans, Louisiana, vol 1 (Long Papers), pp 976–986. https://doi.org/10.18653/v1/N18-1089. https://www.aclweb.org/anthology/N18-1089
40. Zennaki O, Semmar N, Besacier L (2019) A neural approach for inducing multilingual resources and natural language processing tools for low-resource languages. Nat Lang Eng 25(1):43–67
41. Zhang M, Zhang Y, Che W, Liu T (2014) Type-supervised domain adaptation for joint segmentation and pos-tagging. In: Proceedings of the 14th conference of the European chapter of the association for computational linguistics, pp 588–597

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.