Multi-Domain Named Entity Recognition with Genre-Aware and Agnostic Inference

Jing Wang*, **Mayank Kulkarni***, **Daniel Preoţiuc-Pietro**
Bloomberg
New York, New York, USA
{jwang1621,mkulkarni24,dpreotiucpie}@bloomberg.net

Abstract

Named entity recognition is a key component of many text processing pipelines and it is thus essential for this component to be robust to different types of input. However, domain transfer of NER models with data from multiple genres has not been widely studied. To this end, we conduct NER experiments in three predictive setups on data from: a) multiple domains; b) multiple domains where the genre label is unknown at inference time; c) domains not encountered in training. We introduce a new architecture tailored to this task by using shared and private domain parameters and multi-task learning. This consistently outperforms all other baseline and competitive methods on all three experimental setups, with differences ranging between +1.95 to +3.11 average F1 across multiple genres when compared to standard approaches. These results illustrate the challenges that need to be taken into account when building real-world NLP applications that are robust to various types of text and the methods that can help, at least partially, alleviate these issues.

1 Introduction

Accurately identifying named entities and their type in texts is a key processing step for many NLP applications. Named entity recognition (NER) is an important component in several tasks including named entity linking (Cucerzan, 2007), co-reference resolution (Ng and Cardie, 2002), question answering (Krishnamurthy and Mitchell, 2015), relation extraction (Culotta and Sorensen, 2004) and usually sits upstream of analytics such as sentiment (Pang and Lee, 2004) or stance (Mohammad et al., 2016). Building robust NER models to accurately tag and adapt to heterogeneous types of text is thus paramount. Recent research focused on improving the overall performance of NER models on specific data sets. Yet NER models show relatively high variance even when trained on the same data (Reimers and Gurevych, 2017) and poorly generalize when tested on data from different genres[1], especially if these contain entity mentions unseen in the test data (Augenstein et al., 2017; Agarwal et al., 2020).

Despite this, research on NER models robust to different types of input is usually limited to the standard domain adaptation scenario: a single source domain rich in training data and a single target domain with limited or no training data (Lin and Lu, 2018). We argue that this is an over-simplified experimental setup that is not typical for how NER models are used in real-world applications. Ideally, NER models use all available data, regardless of genre, and perform inference on data from any genre, even if this was not encountered in training. In this scenario, simply pooling all the available data is likely sub-optimal as genre-specific differences in named entity mentions are useful to model. Conversely, models limited to only data from the same genre as the test set are likely to underperform, as using more data is usually beneficial.

This work introduces three experimental setups for the NER task where models are trained on data from multiple genres and evaluated as follows:

a) **Multi-Domain** – evaluation is performed across multiple genres, all seen in training.
b) **Multi-Domain with Unknown Domain Labels** – evaluation is carried out across multiple genres, all seen in training, but the genre label for each document is unknown at inference time.
c) **Zero-shot Domain** – evaluation is performed on documents from genres unseen in training.

*Equal Contribution

[1]Throughout this paper, we refer by *genre* to a collection of documents with variations in style or structure that might impact modelling (Santini et al., 2006); we use *domain* when referring to modeling concepts.

We propose a neural architecture for NER tailored to these three experimental setups, based on the popular BiLSTM-CRF architecture (Lample et al., 2016). We augment the base architecture to learn both domain-specific and independent features through shared and private domain components including projections and CRFs. Further, we add a multi-task learning objective for domain prediction to guide this separation. This model can perform inference on a text without knowledge of its corresponding domain label by using the shared components. We compare this model with several competitive methods that use a similar base architecture while holding the embeddings constant (i.e. GloVe embeddings). These include models trained on data from each domain independently, models that pool all data and models that use domain identities as features through to source-target domain adaptation methods.

Extensive results on all three experimental setups on a collection of data from a total of twelve genres demonstrate that our proposed architecture outperforms all others by a respectable margin. Finally, through an error analysis of our results, we aim to understand the contributions of each proposed component and the margins for future improvements.

2 Related Work

Setups for Domain Adaptation Domain adaptation, formulated as learning a single model for the same task across multiple domains, is a well-studied research area in NLP (Chelba and Acero, 2004; Florian et al., 2004; Blitzer et al., 2006; Daumé III, 2007). The standard setup for domain adaptation is to maximize performance on data from a single low-resource (target) domain, by using data from a single high-resource (source) domain (Blitzer et al., 2007; Peng and Dredze, 2017). Extensions consider a single source and multiple different target domains (Yang and Eisenstein, 2015) or multiple sources and a single target domain (Mansour et al., 2009). The multi-domain text classification task studied in (Li and Zong, 2008; Wu and Huang, 2015; Chen and Cardie, 2018) is the analogous setup for the text classification task to the first experimental setup we propose for NER. Under this setup, training and evaluation is done across data from multiple domains.

Multi-Domain Adaptation Methods for multi-domain text classification use data fusion either at the feature or classifier level (Li and Zong, 2008), decomposing the classifier into a shared one and multiple domain-specific ones (Wu and Huang, 2015), further guided by a domain discriminator (Chen and Cardie, 2018) which is also used in multi-lingual NER (Chen et al., 2019). Further, McClosky et al. (2010) explored sequence tagging tasks on data from unknown domains and Chen and Cardie (2018) experiment with sentiment classification on data from unknown domains, similar to our third experimental setup for NER. To the best of our knowledge, our second setup where the domain label is not available at inference time was never explicitly studied. We note that most of these approaches make use of additional unlabeled data from each domain to learn domain-specific representations. We do not use these resources in our methods, as we assume the end-user of the model is agnostic to the data used in training and wants to run inference without having to provide entire comparable corpora.

Domain Adaptation for NER Models for domain adaptation in NER using neural architectures were studied recently, albeit mostly for covering the single-source and single-target setup. The INIT method trains a model using the source domain data, and its parameters are used to initialize a target model which is fine-tuned on the target data (Mou et al., 2016). The MULT method trains jointly one model for each domain with shared parameters (Lee et al., 2018). For sequence tagging, one CRF for each of the two domains is used to obtain the predictions (Yang et al., 2017). Adaptation can also be made at the embeddings stage (Lin and Lu, 2018) or by using additional unlabeled data from the source domain and out-of-domain annotated data (He and Sun, 2017). However, as mentioned above, this assumes that unlabeled training data can be provided for each domain, which may not be realistic. The model adds layers between embeddings and the BiLSTM layers, between the BiLSTM and the CRF for the target domain and separate CRF layers, the latter two of which we adapt to our proposed architecture for multi-domain adaptation. A hierarchical Bayesian prior approach is used in (Finkel and Manning, 2009) to tie feature weights across domains when information is sparse and also allow the model to take advantage if substantial data is available in one domain. Their experiments on NER focused only on three data sets: CoNLL, MUC-6 and MUC-7 and only the first of our three setups. A multi-task domain adaptation

method for NER and word segmentation is used in (Peng and Dredze, 2017). The proposed architecture learns a shared representation across domains and experiments with linear domain projections for each domain to guide learning of shared representations. The output of these linear layers is fed to a CRF. We adopt the linear domain projection method, but extend this to also include a shared projection, followed by domain-specific CRFs and multi-task learning. Finally, another type of domain adaptation is temporal adaptation of models tested on data that is more recent than the training data, when each temporal slice can be considered as a different domain (Rijwhani and Preoţiuc-Pietro, 2020).

3 Methods

This section describes the proposed NER architecture tailored the architecture to our multi-domain experimental setups, which is independent of input embedding representation.

3.1 Base Architecture

The basic component of our NER models is an architecture which has reached state-of-the-art performance several times over the last few years (Lample et al., 2016; Peters et al., 2018; Akbik et al., 2018). Named entity recognition task is a structured prediction task and earlier statistical approaches are based models like Conditional Random Fields (Lafferty et al., 2001), which rely on features often designed based on domain-specific knowledge (Luo et al., 2015). The current dominant approach to the NER task consists of neural architectures based on recurrent neural networks with different choices of input representations (Huang et al., 2015; Ma and Hovy, 2016; Lample et al., 2016; Peters et al., 2018; Akbik et al., 2018, 2019).

The input consists of a concatenation of pre-trained word embeddings and character embeddings. Character embeddings are trained using an LSTM from randomly initialized vectors as in (Lample et al., 2016). Word embeddings are derived from a combination GloVe (Pennington et al., 2014) and FastText (Bojanowski et al., 2017) pre-trained word embeddings, as used in (Ma and Hovy, 2016). The choice of embeddings is orthogonal to the architecture and thus, we hold these constant in all experiments.

This representation is passed through two LSTM layers that process the input sequence in different directions (Huang et al., 2015). The outputs of these layers are concatenated and, in order to map the word representation obtained from the LSTM module into the label distribution, passed to a one-layer feed-forward network. A Conditional Random Field is applied to the class predictions to jointly assign the sequence tags using a transition matrix. This CRF layer improves performance of the model (Lample et al., 2016) as it ensures the output sequence takes into account dependencies between the tags and also models the constraints the output sequence adheres to (e.g. I-PER can not follow B-LOC).

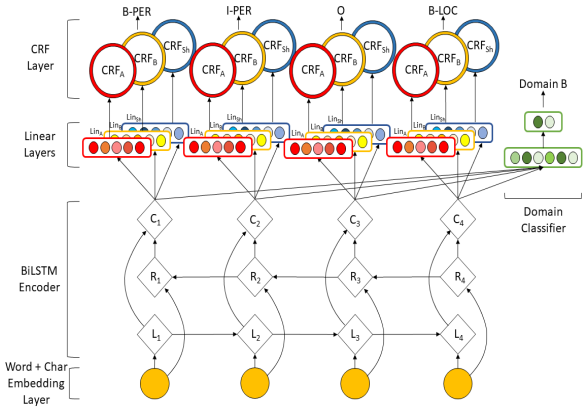

Figure 1: MultDomain–SP–Aux Architecture for 2 domains (A & B) and shared layers denoted by Sh

3.2 Proposed Architecture (MultDomain–SP–Aux)

We propose a new architecture based on the BiLSTM–CRF model tailored to the three proposed experimental setups. Our proposed architecture enhances the base architecture with three components: a) domain -specific and -independent feed-forward layers that process the BiLSTM outputs; b) domain -specific and -independent feed forward layers CRFs; c) a multi-task learning objective that learns domain labels as an auxiliary task.

The proposed architecture changes are motivated by the aim of capturing commonalities in which named entities are referred to, in any given genre, while still allowing for the model to tease apart and exploit domain-specific aspects. The architecture is also designed to capture these commonalities across label relationships, which can vary across domains. In addition, the multi-task objective further assists the model to leverage domain-dependent and -independent components. The choice of input representation is orthogonal to the proposed architecture and our extensions to the architecture can be combined with any input repre-

sentation.

The model architecture is presented in Figure 1 and described below:

Private and Shared Layers We rely on the shared-private paradigm where the model learns both a shared representation across all domains and is useful when the domain of the input is unknown or unseen in training, and a private domain representation that mostly helps tagging in that domain.

We model the shared and private features at both the feature mapping stage connecting the BiLSTM outputs to the CRF(s) and at the CRF level. We expect the features extracted by the BiLSTM layers to model the structure of the input across all domains. The feed-forward layers capture the domain-specific and -independent information by using private output layers for each domain and one shared output layer. In training, the BiLSTM outputs are projected to both the shared layer and the private layer based on the domain label provided in training. The CRF layer is used to make a global decision for the entire tag sequence by modelling label dependencies. We expect that this decision is, at least partially, dependent on domain-specific relationships in the label space. Hence, each feed-forward layer feeds into either private CRFs (one for each domain) or a shared CRF. The separation of the shared and private layers could happen before the CRF stage (late separation) or before the feed-forward layer stage (early separation). We investigate the influence of each individual addition on the multi-domain performance in our analysis section through ablation studies.

Given an input, both the shared and the private parameters are used in learning to predict the output. The set of private parameters for each domain are only updated by data from the same domain while the set of shared parameters are updated in a pooled way by taking all available data points in the training stage regardless of the domain characteristics. For a given data point, inference can be run either by: a) passing it though the private components if the domain label is known; b) through the shared components if the domain label in unknown or the domain of the data is unseen in training. To this end, the objective function for the private and shared layers is:

$$\mathbf{L}_{NER\_SP}(x, y) = \mathbf{L}_{NER\_S}(x, y) + \mathbf{L}_{NER\_P}(x, y) \quad (1)$$

where $\mathbf{L}_{NER\_S}$ and $\mathbf{L}_{NER\_P}$ stand for the shared layer loss and private layer loss respectively.

Multi-Task Learning of Domain Labels Further, to better guide the learning process, we augment our architecture with a multi-task learning objective. Through this, the model learns to predict the domain label of each sample in training as an auxiliary task. The architecture uses average pooling on BiLSTM outputs followed by a fully connected layer. Finally, softmax is applied over the learned domain feature to obtain a probability distribution of all domain labels. The domain classification objective is to minimize the cross-entropy loss $\mathbf{L}_{domain}(x, y_d)$ for an input $\mathbf{x}$ with domain label y_d. The global objective function is the combination of the NER loss function and domain loss:

$$\mathbf{L}(x; y, y_d) = \mathbf{L}_{NER\_SP}(x, y) + \mathbf{L}_{domain}(x, y_d) \quad (2)$$

4 Experimental setup

4.1 Data

We use a collection of data sets spanning eight genres to evaluate our methods. In addition, in order to test the feasibility of NER tagging in a zero-shot domain setup, we present additional data covering four other genres. Each genre of documents is considered a domain in modelling.

4.1.1 Data Sets

The data set collection used in learning the multi-domain models (denoted as 'Open Data' in the rest of the paper) includes the following three data sets:

CoNLL 2003 We use the data set released as part of CoNLL 2003 shared task for English (Tjong Kim Sang and De Meulder, 2003), which is arguably the most popular data set for NER and is regularly used as a benchmark for this task. This data is a collection of news articles from the Reuters Corpus.

Twitter The Twitter data set consists of 22,000 tweets representative of multiple English-speaking locales and a variety of topics that span 11 years of Twitter posts (2009–2019). This data was annotated with Organizations (ORG), Persons (PER) and Locations (LOC), using the annotation guidelines used in annotating past data sets (Tjong Kim Sang and De Meulder, 2003) supplemented with examples that are specific to Twitter data.

OntoNotes (six genres) The OntoNotes data set (Hovy et al., 2006) consists for six different genres annotated, amongst others, with named entities and their types. In this data, each genre refers to a different source, which includes newswire (NW),

Data Set	# Tokens	Density	Entity Distribution		
			ORG	PER	LOC
CoNLL 2003	302811	14.52%	33.2%	38.8%	28.0%
Twitter	227019	8.02%	36.9%	46.5%	16.5%
OntoNotes-NW	490738	8.89%	55.1%	21.1%	23.8%
OntoNotes-BN	258625	9.06%	27.5%	37.2%	35.3%
OntoNotes-MZ	197520	7.84%	28.1%	41.9%	30.0%
OntoNotes-BC	239236	5.49%	27.5%	39.8%	32.8%
OntoNotes-TC	114463	1.59%	12.3%	45.6%	42.1%
OntoNotes-WB	490738	2.17%	25.5%	44.4%	30.1%
Zero-Shot-A	103992	3.10%	53.3%	24.4%	22.2%
Zero-Shot-B	794199	8.48%	55.5%	28.4%	16.1%
Zero-Shot-C	156032	10.06%	64.4%	14.4%	21.1%
Zero-Shot-D	27522	5.84%	38.8%	31.9%	29.4%

Table 1: Size of data sets, NE density (tokens that are named entities) and distributions across entity types for both open and zero-shot data sets.

broadcast news (BN), broadcast conversation (BC), magazine (MZ), telephone conversation (TC) and web data (WB) (Pradhan et al., 2013). Note that we replace the 'LOC', 'FAC' and 'GPE' tags in the OntoNotes data with the 'LOC' type in order to be consistent with the definition of 'LOC' in CoNLL 2003, as also done in (Augenstein et al., 2017).

Zero Shot Genres Finally, for zero-shot genre NER, we use a collection of internal data sets from four different genres spanning news, closed captions and other documents. All four genres were annotated with the same entity types and using similar guidelines.

4.1.2 Data Set Statistics

Data set statistics are presented in Table 1. This shows that all domains are represented with a substantial number of sentences, although the prevalence of named entities and their distribution across types varies, as expected from data sets collected from different sources and genres. We also see that the zero-shot domains are significantly different in entity type distribution and density than the training data, making them well-suited for this setting.

4.1.3 Data Processing

In order to present comparable results across all different data sets, we limit our experiments to three different types of entities that are present in all the above data sets and annotated using similar guidelines: organizations (including geo-political entities and facilities), persons and locations. In case other types of entities exist in the data (e.g. MISC for CoNLL, dates for OntoNotes), these are considered to be not an entity, similar to (Augenstein et al., 2017).

We used the BIO tagging scheme in all our experiments, as this is arguably the most popular and differences in results between this tagging scheme and others, such as the BILOU scheme, are very small in practice (Ratinov and Roth, 2009).

4.1.4 Data Splits

We train our models using the open data sets from CoNLL, Twitter and OntoNotes. The training, development and test splits of CoNLL and OntoNotes follows the standard splits. Similarly, we randomly split the Twitter data set randomly into 70% for training, 10% for development and 20% for testing. The final train, dev and test sets are obtained by joining all the respective splits across the individual data sets.

4.2 Other Methods

We evaluate several baseline methods and other competitive methods introduced in past research and compare to our proposed architecture (**MultDomain–SP–Aux**) described in Section 3.2. These methods focus on different variations of the neural model architecture, while holding the input embeddings constant.

InDomain trains an individual NER model using the base architecture for each of the known domains. In inference, the corresponding in-domain model is used. This allows us to establish the baseline individual domain performance when no information is shared between the domains in training.

InDomain-DomainClassifier uses the same NER models as the InDomain model. The InDomain approach is however unable to directly perform inference on sentences where the domain label is unknown at inference time. We thus build a separate domain classifier using a Bi-LSTM recurrent neural network that feeds the final hidden state into a feed-forward network to recognize the domain of a given input sentence and route it to the appropriate InDomain NER model.

PoolDomain naively pools all available data, disregarding the domain information and trains a model using the base architecture. This model thus ignores the domain information when training, albeit uses all available training data. Data pooling is the standard baseline in most domain adaptation experiments.

PoolDomain-Init uses all available data and uses the domain information to train models on data from one domain at once. After training on data from each domain, the model uses the weights as

initialization for training on next domain. This is similar to the INIT strategy for domain adaptation used in (Mou et al., 2016; Lee et al., 2018). We perform this weight initialization and fine-tuning process over all the domains consecutively, where the order is defined by the density of entities, starting with the highest one.

PoolDomain-GradRev trains the base architecture using a gradient reversal layer (Ganin and Lempitsky, 2014). The gradient reversal technique aims to confuse the domain discriminator while learning NER with the combination of the training data from all domains.

PoolDomain+DomainFeat trains a base architecture model over all available data and, in addition to the text-based features, the domain information is explicitly represented by passing it through a domain embedding. This is appended to the word-level features that are used as input to the BiLSTM layers. The domain embeddings are randomly initialized.

MultDomain-SP extends the MULT method (Yang et al., 2017) to the multi-domain setup. This method uses a domain-specific CRF for each domain and a shared CRF for all domains. Both the BiLSTM and the feed-forward layers are shared across all domains. Inference can be done either through the private layer corresponding to the domain of the input – denoted as **MultDomain-MultCRF (P)** – or through the shared layer – denoted as **MultDomain-MultCRF (S)** – in which case this can be used when the domain label is unknown in inference.

4.3 Implementation Details

For our experiments, we largely follow the training and evaluation procedure used in (Akbik et al., 2018). As hyperparameters, we follow most suggestions outlined in the in-depth study on model robustness (Reimers and Gurevych, 2017). Our training uses 256 hidden states for BiLSTM with mini-batch size of 32. The model parameters are updated using back-propagation and Adam optimizer (Kingma and Ba, 2014). The learning rate is $1e^{-3}$ with weight decay value $1e^{-5}$. The model is regularized with a locked dropout rate of 0.5. We use 300-dimensional pre-trained word embeddings as described in Section 3.1, whereas the character LSTM is randomly initialized and has a hidden dimension of 64. The embeddings are updated on the training data. When training the domain features together with the NER (**PoolDomain+DomainFeat**), we set the domain embedding size to 128. We train all models for 20 epochs and report the results for the model performing best on the joint development set of the open data set collection.

5 Results

In this section, we present and compare the results of all the methods introduced previously. Experiments are conducted first on the open data collection introduced in Section 4.1 in the Multi-Domain and Multi-Domain with Unknown Label setups. Following, we evaluate the performance of our model on the data used for zero-shot genre NER.

The goal of these experiments is to examine the NER performance across the three proposed experimental setups which focus on model generalizability across multiple domains. We note that the results below can not be directly compared to the state-of-the-art results on each data set, as we restrict the entity types to PER, ORG, LOC, such that these types are constant across all data sets.

5.1 Multi-Domain with Known Domain Labels

First, we compare models when assuming the domain label of each test document is known at inference time. The results are listed in Table 2.

Our proposed method – **MultDomain-SP-Aux (P)** – obtains the best results across the entire test collection in both micro-average (+0.43) and macro-average (+1.94) compared to all other approaches and performs best on 7 out of the 8 domains. The second best method is the **PoolDomain+DomainFeat** which uses the domain feature as input. Our method consistently surpasses the in-domain classifiers (**InDomain**) on micro-average (+1.48) and macro-average (+3.11), showing the limitations of naive modeling approaches. Although increases exist across all domains, these are most prominent in domains like TC (+5.36) that have a low density of named entities and where in-domain models have access to limited amounts of data. However, the in-domain performance is better than the pooled method of training, which shows consistent drops in performance on some domains (-8.69 on WB, -6.77 on BC, - 1.98 on CoNLL), where information from other domains did not benefit the model.

Model	Works on Unknown Domain Labels	CoNLL	Twitter	NW	BN	MZ	BC	TC	WB	μ–Avg	M–Avg
InDomain	✗	89.91	67.36	**91.09**	91.09	86.90	84.41	77.06	64.74	85.29	81.57
InDomain+DomainClassifier	✓	*88.92*	66.98	90.48	90.21	85.63	84.64	76.28	59.62	83.93	80.35
PoolDomain	✓	87.93	66.21	90.86	92.76	87.73	89.06	70.29	56.05	83.94	80.11
PoolDomain–Init	✓	31.31	15.74	63.34	67.63	47.30	63.30	33.93	57.55	47.00	47.55
PoolDomain–GradRev	✓	83.49	54.55	83.95	86.87	77.46	83.93	77.78	50.88	77.29	74.86
PoolDomain+DomainFeat	✗	**90.74**	67.80	90.32	92.27	89.12	89.86	78.40	63.37	86.34	82.74
MultDomain–SP (P)	✗	87.70	59.16	88.96	93.51	88.52	89.95	77.97	55.51	82.12	80.16
MultDomain–SP (S)	✓	87.41	57.98	88.64	93.47	88.39	89.00	55.51	54.39	81.73	80.08
MultDomain–SP–Aux (P)	✗	90.21	**69.15**	**91.09**	**93.64**	**91.38**	**90.67**	**82.42**	**67.44**	**86.77**	**84.68**
MultDomain–SP–Aux (S)	✓	88.43	67.13	91.26	93.59	87.67	89.54	78.77	59.63	84.68	82.30

Table 2: Experimental results on the eight data sets, as well as micro (μ-) and macro (M-) averaged across data sets. Performance is measured using micro F1 score. The rows with ✓ indicate methods that can be applied when the domain label is not known at inference time. (S) and (P) denote if inference is done through the shared (S) or private (P) layers of the architecture. Results in bold are the best across all models, those underlined are best across methods that work with unknown domain labels.

5.2 Multi-Domain with Unknown Domain Labels

We now focus on the experimental setup where domain labels are unknown for each data point at inference time. This is akin to a setup where the user is agnostic to the data the model was trained on. As only a subset of the models can perform inference in this scenario, the results are a subset of those in Table 2.

Our model – **MultDomain-SP-Aux (S)** – gains the best overall performance in this setup, with 1.95 macro-average F1 increase over the next best method (**InDomain+DomainClassifier**). The other standard baseline for domain adaptation (**PoolDomain**) obtains a similar performance (-2.19 compared to our method) to the in-domain approach, which shows the benefits of multi-domain adaptation.

PoolDomain-Init is performing overall poorly, which shows that the INIT transfer learning strategy that is somewhat effective for source-target domain adaptation does not work well in the multi-domain setup. Our intuition is that this technique is unable to learn robust features sequentially across N domains, as it performs poorly on the initial trained domains. **PoolDomain-GradRev** gains relatively weak performance overall, lower than the in-domain baseline.

5.3 Zero-Shot Domain

Finally, we show the results on the experimental setup where the test data is the four 'Zero-Shot Genres', which were not used in during training. Table 3 shows the experimental results of all methods that can run inference with unknown domain labels, as we assume that in this setup, the end-user does not have knowledge about the domains used in training and which of these are most similar to the test point.

Models	Zero-Shot Genres				M–Avg
	A	B	C	D	
InDomain+DomainClassifier	47.16	60.04	62.00	59.50	57.17
PoolDomain	52.61	62.53	63.53	61.55	60.05
PoolDomain-Init	24.38	36.92	47.13	19.47	31.98
PoolDomain-GradRev	49.48	68.97	67.95	57.41	60.95
MultDomain-SP (S)	50.9	**72.27**	68.19	61.86	63.30
MultDomain-SP-Aux (S)	**54.50**	67.77	**70.30**	**64.02**	**64.15**

Table 3: Evaluation results on data from genres unseen in training.

Results show that our proposed method obtains again the best results, with a consistent margin of 2.24 macro-average F1 improvement over the next method. Pooling all data (**PoolDomain**) obtains better performance than building in-domain classifiers with domain classification (**InDomain+DomainClassifier**) unlike in the other setups. This also shows that the zero-shot domains we used are indeed different to any of the ones in training and pooling all data manages to build a slightly more robust model than individual ones trained on less data. The in-domain models perform 5.21 F1 points lower than our approach, the largest gap in all experimental setups, highlighting the robustness of the multi-domain modeling approach. The **MultDomain-SP (S)** model is second best, and as this is the base for our method, we discuss its performance in the ablation study from the next section.

6 Analysis

6.1 Ablation Experiments

We first focus on understanding the impact of each component added to our proposed method over the base architecture through an ablation study. Table 4 shows results using the private layer (**MultDomain-SP-Aux (P)**) when each of the three components are alternatively turned off: Shared-Private Linear layer, Shared-Private CRF and the domain prediction auxiliary task.

Shared vs. Shared-Private CRF With the rest of the architecture fixed, the results show that the shared-private CRF performs close to the shared CRF when the shared linear layer is used (80.08 vs. 80.16; 82.04 vs. 82.74; all comparisons in this section are on macro-average). However, once we use a separate linear layer between the BiLSTM and each CRF, the difference between having the shared and the shared-private CRFs increases drastically (81.36 vs. 83.11; 82.30 vs. 84.68). With only this late separation, the inputs to CRF decoders are still domain-independent features, which makes it hard for the linear CRF to adapt. When the inputs are already domain-dependent, the linear CRF can better use this information in performing the joint inference of the sequence. We note that only using shared-private CRF with the base architecture is equivalent to the **MultDomain-SP** method (Yang et al., 2017).

Shared vs. Shared-Private Linear Projections The results show that regardless of the other parameters, adding shared and private linear layers between the BiLSTM layers and the CRF(s) is always beneficial (80.08 vs. 81.36; 80.16 vs. 83.11; 82.04 vs. 82.30; 82.74 vs. 84.68). The improvements are relatively larger when combined with shared and private CRF, as previously seen.

Multi-Task Learning of Domain Labels Finally, we compare the impact of adding the multi-task learning objective. We find that, similar to the linear layers, adding the domain prediction task is always beneficial for the model with the increase being larger if is only a shared linear layer.

We expect that the two tasks at different levels of granularity rely on shared structure in the original semantic space. The document-level domain labels can help regularize the training, providing generic information about which low-level features are valuable to entity-level recognition.

6.2 InDomain with Oracle Choice

In order to understand the limitations of the multi-domain setup, we study whether the models we can build from the available data could theoretically achieve better overall performance. We use an oracle-based selection technique on the in-domain models to select, **after the prediction and using the gold labels** the model which performed best for each test instance, as selected using F1 score or, if there are no entities, the model with most O predictions. If multiple models are tied, we choose one at random. The oracle thus provides the counterfactually "Optimal" strategy of model selection for each test instance and represents an upper bound on strategies relying on InDomain models.

Table 5 compares the oracle strategy predictions with the **InDomain+DomainClassifier** and the **MultDomain-SP-Aux** model. The results show that even though our model improves substantially over the in-domain models, an oracle selection method would push performance much higher (+6.73 F1 on the open data). This highlights both the variability of NER models trained on different data sets and that there is potentially more room for improvements in the multi-domain setup.

6.3 InDomain Models

The Supplementary Material shows a breakdown of the domain prediction labels for three methods: domain classification, domain prediction in the proposed **MultDomain-SP-Aux** model and the oracle in-domain choice on gold data. The oracle strategy selects the predictions from all in-domain models. Based on this, we analyzed the performance of each individual in-domain model when tested on all domains in Table 6. We find that although the Oracle strategy uses a mix of models, any model alone is unable to generalize to other domains (67.19 vs. 84.68 best InDomain model compared to the best overall model). In the zero-shot genres, the Twitter model performs close to the **MultDomain-SP-Aux** model (-0.56 F1), albeit it is 24 F1 lower on the multi-domain setup. This reinforces that learning shared domain features as opposed to learning individual models helps boost performance and is more robust to different types of inputs.

7 Runtime Comparison

Finally, we compare the runtime difference across various methods listed in the experiment section to test the practical implications of using our pro-

Auxiliary Task	Linear	CRF	CoNLL	Twitter	NW	BN	MZ	BC	TC	WB	μ–Avg	M–Avg
✗	Shared	Shared	87.41	57.98	88.64	93.47	88.39	89.00	55.51	54.39	81.73	80.08
		Sh-Private	87.70	59.16	88.96	93.51	88.52	89.95	77.97	55.51	82.12	80.16
✗	Sh-Private	Shared	87.65	64.45	90.88	92.82	87.92	88.75	80.60	57.81	83.77	81.36
		Sh-Private	89.57	67.78	90.98	92.45	90.10	88.75	80.86	64.38	85.95	83.11
✓	Shared	Shared	89.00	67.27	91.10	93.00	89.15	89.00	78.36	59.48	85.69	82.04
		Sh-Private	89.48	67.19	**91.31**	93.48	89.99	89.48	79.18	61.84	86.55	82.74
✓	Sh-Private	Shared	88.43	67.13	91.26	93.59	87.67	89.54	78.77	59.63	84.68	82.30
		Sh-Private	**90.21**	**69.15**	91.09	**93.64**	**91.38**	**90.67**	**82.42**	**67.44**	**86.77**	**84.68**

Table 4: Ablation study comparing the performance (F1 score) of models trained with and without: shared-private linear projections of BiLSTM outputs, shared-private CRF heads and multi-task domain classification.

Model	Open Data	Zero-Shot
InDomain + DomainClassifier	80.35	57.17
MultDomain-SP-Aux	84.68	64.15
Oracle with InDomain	91.41	80.27

Table 5: Performance in macro-average F1 of the InDomain models with an oracle model selection strategy using gold test data compared to selected methods.

Model	Open Data	Zero-Shot
CoNLL	64.26	61.40
Twitter	60.59	63.59
NW	67.19	59.00
BN	66.08	54.82
MZ	57.52	48.62
BC	59.19	46.30
TC	47.25	37.41
WB	44.09	25.41

Table 6: Results of InDomain models trained on each domain independently on the open data set collection and the zero-shot genres reported in macro average of F1 for each domain.

posed multi-domain modelling approach. In test phase, we set the batch size as 128. Table 7 shows the average time of inference time used for each model. Our proposed model architecture takes 0.15 ms (33% increase) longer for inference than **InDomain** or **PoolDomain** models, which is a result of more model parameters. However, our proposed architecture is still 0.19 ms faster than using the **InDomain+DomainClassifier** approach.

In addition to inference runtime, we also find that the training time is not significantly more than the combined training time of N in-domain models. The main additions are that of the shared layers and the auxiliary task to the components of the N in-domain models and is thus a constant addition in the number of parameters to the total of N in-domain models. Hence, the model would scale by a constant with respect to the number of input domains (N+1 number of components, where N is the number of domains). This should allow our proposed model to scale to a large number of domains.

This highlights that the proposed **MultDomain–SP–Aux** model is a viable option for real-world applications.

Model	Runtime (ms)
InDomain	0.45
InDomain+DomainClassifier	0.79
PoolDomain	0.45
PoolDomain–Init	0.43
PoolDomain–GradRev	0.47
PoolDomain+DomainFeat	0.45
MultDomain–SP	0.56
MultDomain–SP–Aux	0.60

Table 7: Averaged inference time (in ms) per sentence query on Open Dataset.

8 Conclusions

Robustness of NLP models is essential to their wider adoption and usability. Existing NER approaches are widely faced with limited scalability when applied to data that spans multiple domains. This paper introduced three experimental setups that provide a framework for evaluating the robustness of NER models. These include learning from data in multiple domains and testing on all domains, when the domain label of the test point is unknown and when this does not belong to a domain seen in training. Building on past research, we proposed a new neural architecture that achieves substantial improvements of up to 5 F1 points when compared to standard methods. Future work will focus on domain adaptation at the embedding layer.

References

Oshin Agarwal, Yinfei Yang, Byron C. Wallace, and Ani Nenkova. 2020. Interpretability analysis for named entity recognition to understand sys-

tem predictions and how they can improve. *ArXiv*, abs/2004.04564.

Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019. Pooled contextualized embeddings for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 724–728, Minneapolis, Minnesota. Association for Computational Linguistics.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Isabelle Augenstein, Leon Derczynski, and Kalina Bontcheva. 2017. Generalisation in named entity recognition: A quantitative analysis. *Computer Speech & Language*, 44:61–83.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic. Association for Computational Linguistics.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128, Sydney, Australia. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ciprian Chelba and Alex Acero. 2004. Adaptation of maximum entropy capitalizer: Little data can help a lo. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 285–292, Barcelona, Spain. Association for Computational Linguistics.

Xilun Chen, Ahmed Hassan Awadallah, Hany Hassan, Wei Wang, and Claire Cardie. 2019. Multi-source cross-lingual model transfer: Learning what to share. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3098–3112, Florence, Italy. Association for Computational Linguistics.

Xilun Chen and Claire Cardie. 2018. Multinomial adversarial networks for multi-domain text classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1226–1240, New Orleans, Louisiana. Association for Computational Linguistics.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716, Prague, Czech Republic. Association for Computational Linguistics.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 423–429, Barcelona, Spain.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic. Association for Computational Linguistics.

Jenny Rose Finkel and Christopher D Manning. 2009. Hierarchical bayesian domain adaptation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 602–610.

R Florian, H Hassan, A Ittycheriah, H Jing, N Kambhatla, X Luo, N Nicolov, and S Roukos. 2004. A statistical model for multilingual entity detection and tracking. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 1–8, Boston, Massachusetts, USA. Association for Computational Linguistics.

Yaroslav Ganin and Victor Lempitsky. 2014. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*.

Hangfeng He and Xu Sun. 2017. A unified model for cross-domain and semi-supervised named entity recognition in chinese social media. In *Thirty-First AAAI Conference on Artificial Intelligence*, AAAI.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA. Association for Computational Linguistics.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *ArXiv*, abs/1508.01991.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Jayant Krishnamurthy and Tom M. Mitchell. 2015. Learning a compositional semantics for Freebase with an open predicate vocabulary. *Transactions of the Association for Computational Linguistics*, 3:257–270.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML, pages 282–289.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Ji Young Lee, Franck Dernoncourt, and Peter Szolovits. 2018. Transfer learning for named-entity recognition with neural networks. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Shoushan Li and Chengqing Zong. 2008. Multi-domain sentiment classification. In *Proceedings of ACL-08: HLT, Short Papers*, pages 257–260, Columbus, Ohio. Association for Computational Linguistics.

Bill Yuchen Lin and Wei Lu. 2018. Neural adaptation layers for cross-domain named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2012–2022, Brussels, Belgium. Association for Computational Linguistics.

Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. 2015. Joint entity recognition and disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 879–888, Lisbon, Portugal. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 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 (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. 2009. Domain adaptation with multiple sources. In *Advances in Neural Information Processing Systems*, NeurIPS, pages 1041–1048.

David McClosky, Eugene Charniak, and Mark Johnson. 2010. Automatic domain adaptation for parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 28–36. Association for Computational Linguistics.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. SemEval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41, San Diego, California. Association for Computational Linguistics.

Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2016. How transferable are neural networks in NLP applications? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 479–489, Austin, Texas. Association for Computational Linguistics.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 104–111, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain.

Nanyun Peng and Mark Dredze. 2017. Multi-task domain adaptation for sequence tagging. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 91–100, Vancouver, Canada. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark. Association for Computational Linguistics.

Shruti Rijwhani and Daniel Preoţiuc-Pietro. 2020. Temporally-informed analysis of named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Marina Santini, Richard Power, and Roger Evans. 2006. Implementing a characterization of genre for automatic genre identification of web pages. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 699–706, Sydney, Australia. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Fangzhao Wu and Yongfeng Huang. 2015. Collaborative multi-domain sentiment classification. In *2015 IEEE International Conference on Data Mining*, pages 459–468. IEEE.

Yi Yang and Jacob Eisenstein. 2015. Unsupervised multi-domain adaptation with feature embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 672–682, Denver, Colorado. Association for Computational Linguistics.

Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. ICLR.

A Domain Prediction

We further study the domains that are selected by the methods above by creating confusion matrices between the domain predictions of three setups: domain classification, domain prediction in the proposed **MultDomain-SP-Aux** model and the oracle in-domain choice on gold data. Figure 2 shows that the Oracle model relies on the corresponding InDomain model to only a limited extent

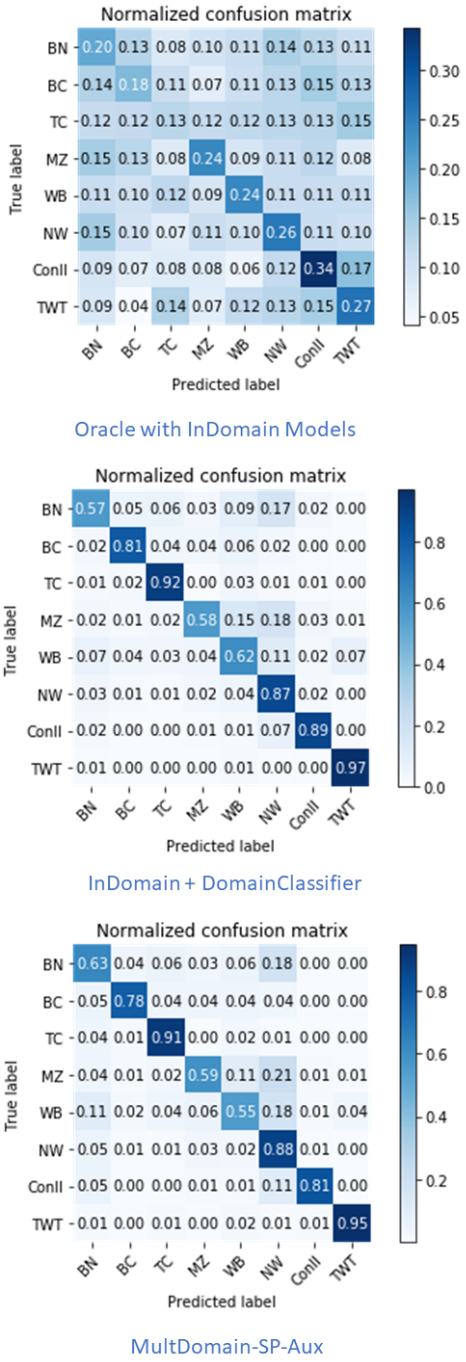

Figure 2: Domain label confusion matrices on the CoNLL-Twitter-OntoNotes data collection.

for each model. In uniformly many cases, predictions from other in-domain models are better than the existing in-domain one, showing the variability of the NER models. The domain classifier predictions align closer to the actual domains. The **MultDomain-SP-Aux** model also tends to predict the domain correctly, but we see that it better learns the NW, WB and BN domains. Note noting that the MultDomain-SP-Aux model does not use these domain predictions in inference and the model uses the shared components for unknown domains or

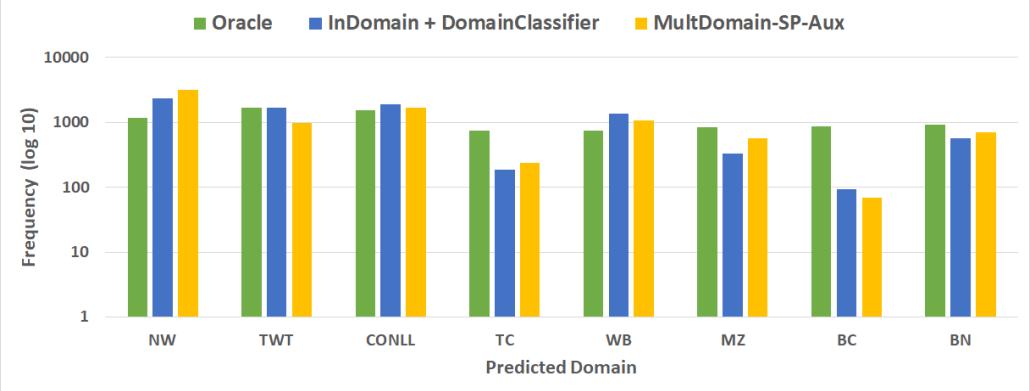

Figure 3: Zero-Shot Domain data domain-label frequency prediction comparison

labels.

Finally, we plot the domain prediction distribution on the zero-shot genre data in Figure 3. We find that similar to the confusion matrices, the oracle strategy has a more even spread in domain selection. We observe similar patterns to the confusion matrices for the **InDomain+DomainClassifier** and **MultDomain-SP-Aux** models.