Hitachi at SemEval-2020 Task 11: An Empirical Study of Pre-Trained Transformer Family for Propaganda Detection

Gaku Morio[*], **Terufumi Morishita**[*], **Hiroaki Ozaki** and **Toshinori Miyoshi**
Hitachi, Ltd.
Research and Development Group
Kokubunji, Tokyo, Japan
{gaku.morio.vn, terufumi.morishita.wp,
hiroaki.ozaki.yu, toshinori.miyoshi.pd}@hitachi.com

Abstract

In this paper, we show our system for SemEval-2020 task 11, where we tackle propaganda span identification (SI) and technique classification (TC). We investigate heterogeneous pre-trained language models (PLMs) such as BERT, GPT-2, XLNet, XLM, RoBERTa, and XLM-RoBERTa for SI and TC fine-tuning, respectively. In large-scale experiments, we found that each of the language models has a characteristic property, and using an ensemble model with them is promising. Finally, the ensemble model was ranked 1st amongst 35 teams for SI and 3rd amongst 31 teams for TC.

1 Introduction

This paper shows our proposed system for the SemEval-2020 task 11: Detection of Propaganda Techniques in News Articles (Da San Martino et al., 2020). The goal of the task was to design a model to detect and classify propaganda. To this end, there are two subtasks: span identification (SI) for predicting propaganda spans and technique classification (TC) for predicting propaganda technique types used for a given span. SI can be assumed as a sequence labeling problem and TC as a multi-label classification problem.

Recent studies such as (Yoosuf and Yang, 2019; Vlad et al., 2019) proposed employing BERT (Devlin et al., 2019), a pre-trained language model, for propaganda detection. Since propaganda detection tasks require highly semantic understanding, leveraging such strong pre-trained language models is promising. However, new state-of-the-art pre-trained language models, such as XLNet (Yang et al., 2019) and RoBERTa (Liu et al., 2019), are being proposed rapidly, and there are no sufficient studies on them in the research on propaganda detection. Revealing the ability of capturing propaganda semantics with state-of-the-art models could contribute to discussing future applications.

Therefore, we investigate state-of-the-art pre-trained language models (`PLMs`) for propaganda detection. We employ not only **BERT** (Devlin et al., 2019) but other `PLMs` such as **GPT-2** (Radford et al., 2019), **RoBERTa** (Liu et al., 2019), **XLM-RoBERTa** (Conneau et al., 2019), **XLNet** (Yang et al., 2019), and **XLM** (Lample and Conneau, 2019). The `PLMs` are fine-tuned by our proposed SI and TC models with various hyperparameters. We also propose an effective ensemble method with stacked generalization, which is generally better than a naive average ensemble.

Our ensemble model was ranked 1st in SI and 3rd in TC, showing that leveraging state-of-the-art `PLMs` is promising. We also empirically gained the following insights as described later.

1. RoBERTa and XLNet generally perform better for propaganda detection.
2. An ensemble model with all types of `PLMs` showed stable and better performance than employing a single `PLM` type.
3. Each `PLM` has a different optimal learning rate, and finding the optimal one is essential to elicit high performance.

[*] Contributed equally.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1739–1748
Barcelona, Spain (Online), December 12, 2020.

2 Related Work

Propaganda generally intends to promote an agenda or point of view with specific information such as biased or misleading descriptions. Since detecting propaganda in text could be useful for further applications such as detecting fake news, research on propaganda detection has been attracting much attention. Barrón-Cedeño et al. (2019) proposed a model for assessing the "level" of propaganda in an article. Da San Martino et al. (2019) focused on more fine-grained analysis, proposing a model for detecting propaganda spans and classifying their techniques. Some recent studies (Yoosuf and Yang, 2019; Vlad et al., 2019; Hua, 2019; Fadel et al., 2019; Tayyar Madabushi et al., 2019) utilized `PLMs` such as BERT (Devlin et al., 2019) to detect fine-grained propaganda. Our work is related to the studies of (Fadel et al., 2019; Al-Omari et al., 2019), which employed ensemble models with `PLMs`. Different from these studies, we further investigate the number of `PLMs`.

3 Pre-Trained Language Models (PLMs)

In this paper, six types of Transformer (Vaswani et al., 2017) based `PLMs` were used. The reason behind this is to unify our implementation and evaluation. We provide a brief description of each `PLM`:

BERT (Devlin et al., 2019) is the epoch-making Transformer-based masked language model. We employ a pre-trained model called *bert-large-cased-whole-word-masking*.

GPT-2 (Radford et al., 2019) is a model that followed Open AI GPT (Radford and Sutskever, 2018). GPT-2 has achieved state-of-the-art results in a zero-shot setting.

RoBERTa (Liu et al., 2019) is a carefully fine-tuned BERT-based model, where the authors investigated hyperparameters and training data size. RoBERTa has achieved state-of-the-art results on major text benchmarks. We employ a pre-trained model called *roberta-large*.

XLM-RoBERTa (Conneau et al., 2019) is a cross-lingual version of RoBERTa. XLM-RoBERTa has outperformed the cross-lingual BERT. We employ a pre-trained model called *xlm-roberta-large*.

XLNet (Yang et al., 2019) employs a generalized autoregressive pre-training in Transformers. XLNet has outperformed BERT in major tasks. We employ a pre-trained model called *xlnet-large-cased*.

XLM (Lample and Conneau, 2019) is a cross-lingual language model, introducing an unsupervised cross-lingual learning. We employ a pre-trained model called *xlm-mlm-en-2048*.

4 Propaganda Detection Models

Given a split sentence[1], our SI model predicts propaganda spans. The TC model in turn predicts propaganda techniques for given spans in a sentence.[2] The technique labels include *Loaded Language*, which uses words with strong emotional implications, *Name Calling or Labelling*, which describe an object or something that the audience fears or sees as undesirable, *Doubt*, which questions the credibility of something, and so on (Da San Martino et al., 2019). Refer to (Da San Martino et al., 2020) for more details. The following subsections give details on the proposed SI and TC models.

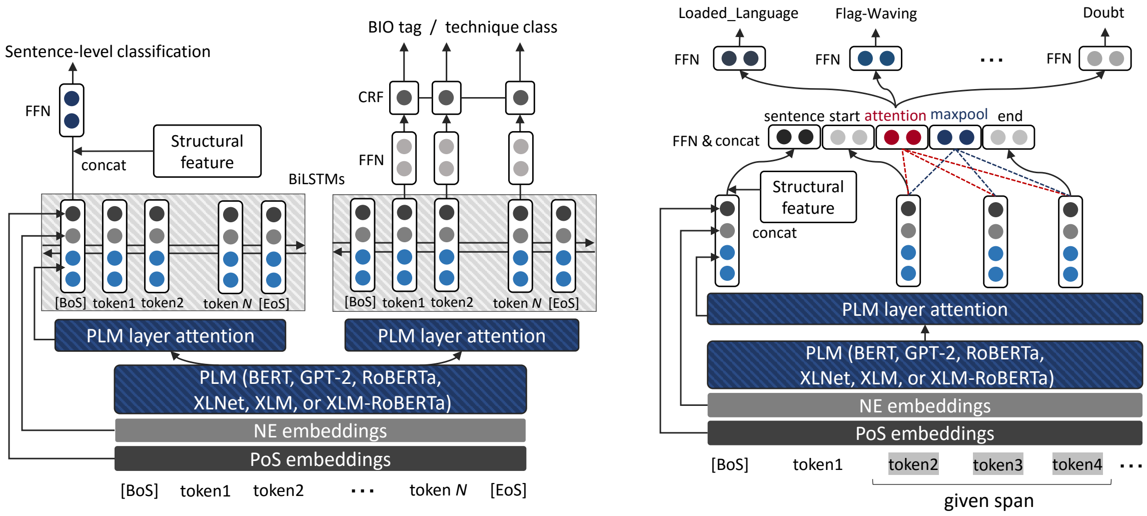
4.1 SI Model

Figure 1a shows an overview of the proposed model for SI. Given a tokenized sentence, $\texttt{PLM} \in \{\texttt{BERT}, \texttt{GPT-2}, \texttt{RoBERTa}, \texttt{XLM-RoBERTa}, \texttt{XLNet}, \texttt{XLM}\}$, and bi-directional long short term memory (BiLSTM) (Graves et al., 2013) encode the sentence, predicting propaganda spans with token-level `BIO` tags, namely propaganda begins (`B`), is inside (`I`), or is outside (`O`) the spans. In addition, we provide two joint auxiliary tasks to effectively train the model. These auxiliary tasks are described later.

Input Representation: To obtain input representations, we provide a layer-wise attention to fuse the

[1] For both SI and TC, given a news article, we split it into sentences. All the training and predictions were conducted on a *sentence level*. Each sentence was tokenized by spaCy (`https://spacy.io/`), jointly obtaining the part-of-speech (PoS) tag and named entity (NE) tag to make common input features.

[2] Note that we consider only sentences that contain propaganda in TC because the propaganda spans are given in the sub-task.


Sentence-level classification
FFN
concat
Structural feature
BIO tag / technique class
CRF
FFN
BiLSTMs
[BoS] token1 token2 token N [EoS]
[BoS] token1 token2 token N [EoS]
PLM layer attention
PLM layer attention
PLM (BERT, GPT-2, RoBERTa, XLNet, XLM, or XLM-RoBERTa)
NE embeddings
PoS embeddings
[BoS] token1 token2 ··· token N [EoS]
Loaded_Language
Flag-Waving
Doubt
FFN
FFN
FFN
···
sentence start attention maxpool end
FFN & concat
Structural feature
concat
PLM layer attention
PLM (BERT, GPT-2, RoBERTa, XLNet, XLM, or XLM-RoBERTa)
NE embeddings
PoS embeddings
[BoS] token1 token2 token3 token4 ···
given span

(a) SI model

(b) TC model

Figure 1: Overview of our proposed models

outputs of $\texttt{PLM}$ layers (Kondratyuk and Straka, 2019; Peters et al., 2018):

$$\mathbf{h}^{(\mathrm{si})}_{\texttt{PLM},i} = c^{(\mathrm{si})} \sum_{j} \texttt{PLM}_{ij} \cdot \mathrm{softmax}\left(\mathbf{s}^{(\mathrm{si})}\right)_j,$$

where $\mathbf{s}^{(\mathrm{si})}$ and $c^{(\mathrm{si})}$ are trainable parameters, and $\texttt{PLM}_{ij}$ is an embedding of the i-th word token in the j-th layer of a $\texttt{PLM}$.[3] We also concatenated part-of-speech (PoS) embeddings ($\mathbf{h}^{(\mathrm{si})}_{\texttt{PoS},i}$) and named entity (NE) embeddings ($\mathbf{h}^{(\mathrm{si})}_{\mathrm{NE},i}$) for each token i.[4] Therefore, the i-th word token is represented as:

$$\mathbf{h}^{(\mathrm{si})}_{i} = \mathbf{h}^{(\mathrm{si})}_{\texttt{PLM},i} \oplus \mathbf{h}^{(\mathrm{si})}_{\texttt{PoS},i} \oplus \mathbf{h}^{(\mathrm{si})}_{\mathrm{NE},i},$$

where $\oplus$ is a concatenate operation.

BiLSTM-CRF: We employed BiLSTM-CRF because we preliminarily found that stacking BiLSTM-CRF (Huang et al., 2015) on a $\texttt{PLM}$ leads to better performance in SI. The input token representations $\mathbf{h}_i$ are fed to the multi-layered BiLSTM to obtain a further contextualized token representation:

$$\mathbf{e}^{(\mathrm{si})}_{i} = \mathrm{BiLSTM}(\mathbf{h}^{(\mathrm{si})}_{i}).$$

We apply a feed forward network (FFN) with one hidden layer and a fully-connected layer to the recurrent states before classification:

$$\hat{\mathbf{y}}^{(\mathrm{si\_bio})}_{i} = W^{(\mathrm{si\_bio})}\mathrm{FFN}^{(\mathrm{si\_bio})}\left(\mathbf{e}^{(\mathrm{si})}_{i}\right) + \mathbf{b}^{(\mathrm{si\_bio})},$$

where $W^{(\mathrm{si\_bio})}$ and $\mathbf{b}^{(\mathrm{si\_bio})}$ are parameters. $\hat{\mathbf{y}}^{(\mathrm{si\_bio})}_{i}$ is the output of $\texttt{B}$, $\texttt{I}$, and $\texttt{O}$ labels.[5] Finally, we employ a conditional random field (CRF) for training and prediction.

[Auxiliary1] Token-Level Technique Classification: This auxiliary task predicts token-level propaganda technique classes to add more information for span identification.[6] In fact, a previous study

[3] For subword-level $\texttt{PLM}$, subwords are averaged, and the averaged output per word token is used as the $\texttt{PLM}_{ij}$.

[4] Special embeddings are used for the PoS and NE embeddings of BoS and EoS tokens.

[5] Note that we ignored special tokens such as the beginning of sentence (BoS) token (e.g., [CLS]) and the end of sentence (EoS) token (e.g., [SEP]) in the classifier.

[6] We extracted TC labels aligning with spans in SI in the training set. We did **not** use any TC labels in the development and test set, and there was no problem when we contacted the organizers.

(Schulz et al., 2018) suggests that joint training with multiple token-level tasks helps improve performance in a low-resource setting. Our expectation is that spans for low-frequency propaganda techniques can be detected with this auxiliary task. We achieve this by simply providing another output layer:

$$\hat{\mathbf{y}}_i^{(\text{si\_tech})} = W^{(\text{si\_tech})}\text{FFN}^{(\text{si\_tech})}\left(\mathbf{e}_i^{(\text{si})}\right) + \mathbf{b}^{(\text{si\_tech})},$$

where $W^{(\text{si\_tech})}$ and $\mathbf{b}^{(\text{si\_tech})}$ are parameters, and $\hat{\mathbf{y}}_i^{(\text{si\_tech})}$ is the output of 14 propaganda technique classes and a non-propaganda class.[7] We also employ a CRF for training and prediction.

[Auxiliary2] Sentence-Level Classification: Given that sentences that contain propaganda are comparatively low in number when compared with non-propaganda sentences, we introduce a sentence-level auxiliary task. This auxiliary task predicts sentence-level classes with lower granularity on the basis of whether the sentence contains propaganda or not, and higher granularity token-level tasks are backpropagated only when a sentence contains propaganda. Therefore, we do not learn much information from non-propaganda sentences for detecting spans. A similar idea was derived from the work of Da San Martino et al. (2019), in which they proposed incorporating a higher granularity task on the basis of lower granularity information (sentence-level task) with a gating mechanism.

We provide another PLM layer attention and multi-layered BiLSTM to distinguish information between sentence-level and token-level tasks:

$$\begin{aligned}\mathbf{h}_i^{(\text{si\_sent})} &= \mathbf{h}_{\texttt{PLM},i}^{(\text{si\_sent})} \oplus \mathbf{h}_{\texttt{PoS},i}^{(\text{si})} \oplus \mathbf{h}_{\texttt{NE},i}^{(\text{si})},\\ \mathbf{e}_i^{(\text{si\_sent})} &= \text{BiLSTM}^{(\text{si\_sent})}(\mathbf{h}_i^{(\text{si\_sent})}).\end{aligned}$$

Finally, we use the output from the BoS token[8] [9], predicting the probability that a sentence contains propaganda:

$$\hat{y}^{(\text{si\_sent})} = \sigma\left(\mathbf{v}^{(\text{si\_sent})\top}\text{FFN}^{(\text{si\_sent})}\left(\mathbf{e}_{\text{BoS}}^{(\text{si\_sent})} \oplus \phi\right) + b^{(\text{si\_sent})}\right),$$

where $\mathbf{v}^{(\text{si\_sent})}$ and $b^{(\text{si\_sent})}$ are parameters, $\mathbf{e}_{\text{BoS}}^{(\text{si\_sent})}$ is the hidden state of the BoS token (e.g., normally $\mathbf{e}_{\text{BoS}}^{(\text{si\_sent})} = \mathbf{e}_1^{(\text{si\_sent})}$), and σ is a sigmoid function. We also concatenated a structural feature vector ϕ including the sentence length and positional information in its article. The positional information includes binary signals if the sentence is located in the upper, middle, or lower.

Objective: Given that we have distinct sentence- and token-level tasks, our objective is described as:

$$\mathcal{L}^{(\text{si})} = \mathcal{L}^{(\text{si\_sent})} + \begin{cases} 0 & \text{if the sentence contains no propaganda spans} \\ \mathcal{L}^{(\text{si\_bio})} + \lambda\mathcal{L}^{(\text{si\_tech})} & \text{else} \end{cases}$$

where $\mathcal{L}^{(\text{si\_sent})}$ is a cross-entropy loss for the sentence-level task, and $\mathcal{L}^{(\text{si\_bio})}$ and $\mathcal{L}^{(\text{si\_tech})}$ are CRF loss with a negative log likelihood for span identification and technique classification, respectively. λ is a hyperparameter for controlling the auxiliary tasks. The objective means that if a sentence contains no propaganda spans, token-level tasks are ignored. Also, we assign a weight to the sentence-level loss according to the inverse proportion of positive samples to deal with class imbalance.

After training, we modify thresholds of the sentence-level task on the basis of validation scores. At inference, if the output probability in the task is below the threshold, it is regarded as a non-propaganda sentence. Propaganda spans are predicted only when the probability is above the threshold.

[7] Given this task is an auxiliary and multi-label samples (i.e., multi-labels in a same span) are much less common, we converted them into a single class and discarded the rest.

[8] To process any type of PLM in the same manner, we inserted original special tokens in the input tokens when no BoS or EoS tokens were available in the PLM.

[9] We accidentally dealt with XLNet encoding as "<cls> tokens <sep>," while the correct form is "tokens <sep> <cls>." The experimental results might had been affected, however, it still showed high performance.

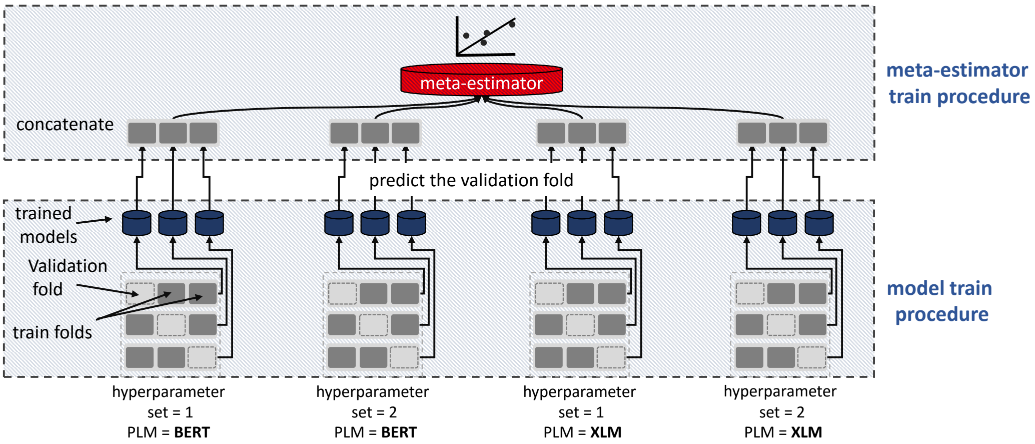

meta-estimator
meta-estimator train procedure
concatenate
predict the validation fold
trained models
Validation fold
train folds
model train procedure
hyperparameter set = 1 PLM = BERT
hyperparameter set = 2 PLM = BERT
hyperparameter set = 1 PLM = XLM
hyperparameter set = 2 PLM = XLM

Figure 2: Example of our ensemble method. We employ stacking and cross-validation. Note that this figure shows only two PLMs and two hyperparameter sets.

4.2 TC Model

Figure 1b shows an overview of the proposed TC model. In TC, given a propaganda span, we predict propaganda technique(s) for each span.[10]

Propaganda Span Representation: To produce a propaganda span representation, we provide two distinct FFNs, feeding input representation $\mathbf{h}_i^{(\mathrm{tc})}$, that were obtained in the same manner as the SI model. One of the two FFNs is for the BoS token and produces sentence representations, and the other is for tokens in a propaganda span:

$$\mathbf{e}_i^{(\mathrm{tc})} = \begin{cases} \mathrm{FFN}^{(\mathrm{tc\_bos})}\left(\mathbf{h}_i^{(\mathrm{tc})} \oplus \phi\right) & \text{if } i \text{ is a BoS token} \\ \mathrm{FFN}^{(\mathrm{tc})}\left(\mathbf{h}_i^{(\mathrm{tc})}\right) & \text{else} \end{cases}$$

where $\mathbf{h}_{\mathrm{BoS}}^{(\mathrm{tc})}$ is a sentence representation obtained from the BoS token. The propaganda span representation is obtained by concatenating the representation of the BoS token ($\mathbf{e}_{\mathrm{BoS}}^{(\mathrm{tc})}$), tokens located at span start ($\mathbf{e}_{\mathrm{start}}^{(\mathrm{tc})}$) and end ($\mathbf{e}_{\mathrm{end}}^{(\mathrm{tc})}$), and representations aggregated by attention ($\mathbf{e}_{\mathrm{att}}^{(\mathrm{tc})}$) and maxpooling ($\mathbf{e}_{\mathrm{maxp}}^{(\mathrm{tc})}$) in the span as follows.

$$\mathbf{e}^{(\mathrm{tc\_span})} = \mathbf{e}_{\mathrm{BoS}}^{(\mathrm{tc})} \oplus \mathbf{e}_{\mathrm{start}}^{(\mathrm{tc})} \oplus \mathbf{e}_{\mathrm{end}}^{(\mathrm{tc})} \oplus \mathbf{e}_{\mathrm{att}}^{(\mathrm{tc})} \oplus \mathbf{e}_{\mathrm{maxp}}^{(\mathrm{tc})},$$

Classifier and Objective: We provide an additional label-wise FFN and linear layer to extract label-specific information for each propaganda technique before prediction:

$$\hat{y}_\ell^{(\mathrm{tc})} = \sigma\left(\mathbf{v}_\ell^{(\mathrm{tc})\top}\mathrm{FFN}_\ell^{(\mathrm{tc})}\left(\mathbf{e}^{(\mathrm{tc\_span})}\right) + b_\ell^{(\mathrm{tc})}\right),$$

where $\mathbf{v}_\ell^{(\mathrm{tc})}$ and $b_\ell^{(\mathrm{tc})}$ are trainable parameters, and ℓ denotes a technique label such as *flag-waving*. Since TC is a multi-label problem, we provide a binary cross-entropy loss for each class. Similar to Da San Martino et al. (2019), we assign weight to a loss according to the proportion of positive samples to deal with class imbalance. After training, we multiply the output probability for each label on the basis of the validation scores automatically. At inference, we sort predicted labels for each sentence in descending order and assign labels according to the order in a multi-label span.[11]

[10] There are 14 possible labels: *appeal to authority*, *appeal to fear-prejudice*, *bandwagon, reductio ad hitlerum*, *black-and-white fallacy*, *causal oversimplification*, *doubt*, *exaggeration, minimization*, *flag-waving*, *loaded language*, *name calling, labeling*, *repetition*, *slogans*, *thought-terminating clichés*, or *whataboutism, straw men, red herring*. Refer to the task description paper (Da San Martino et al., 2020) for more details.

[11] This is because the number of labels in a span was given in TC.

4.3 Ensemble with Stacking

We propose an ensemble strategy based on the concept of stacked generalization (Wolpert, 1992). Stacked generalization feeds prediction results (i.e., output probabilities from classifiers) into a meta-estimator and trains the estimator with gold labels. In this study, the keys are hyperparameter search and cross-validation. The simplified training procedure of the ensemble model can be found in Figure 2.

In the procedure for model training, for either the SI and TC, assume we have k-fold cross-validation, N_H hyperparameter sets, and N_P PLMs. A hyperparameter set includes the dropout ratio and learning rate. For each hyperparameter set, we fine-tune the SI or TC models with training folds without using the validation fold. Therefore, $N_P \times k$ models for each hyperparameter set are generated. To select better models, we use only the top N_{HT} hyperparameter sets on the basis of the validation score, resulting in $N_{HT} \times N_P \times k$ models. For example, as in Figure 2, $N_{HT} = 2$, $N_P = 2$ (i.e., BERT and XLM), and $k = 3$.

In the meta-estimator training procedure, we train a linear model (i.e., classifier or regressor) on the basis of the outputs of the fine-tuned SI or TC models. First, we predict the validation fold in the training data through the fine-tuned model. By concatenating the predicted validation folds for each hyperparameter set, the representative *out-of-folds* of each hyperparameter set are organized. This means that we have meta-features $D \in \mathbb{R}^{d \times N_{HT}}$ to train the meta-estimator, where d is the size of the training data.

In the test procedure, we predict test labels with the fine-tuned models in the top hyperparameter sets that were selected in the training step. The predicted labels are then fed into the trained meta-estimator, obtaining final predictions.

Meta-Estimators: We employ the ridge classifier (Hoerl and Kennard, 1970) implemented in scikit-learn (Pedregosa et al., 2011) for the meta-estimator. We estimate that even with a naive linear model, the final outputs are generally more robust and accurate. We provided the meta-estimators for the sentence-level task in SI, BIO classification in SI, and all TC labels. The meta-estimators of the sentence-level task and TC labels receive the output probabilities of the corresponding labels as input representations.

5 Experiments

Implementation Detail: All SI and TC models were implemented with PyTorch (Paszke et al., 2019) and Hugging Face's transformer library (Wolf et al., 2019). Layer attentions were applied for the last eight layers in all PLMs, employing dropout. CRF classifiers were implemented using pytorch-crf [12]. At training, we split the network into two parameter groups: one for the parameters of PLM and one for all other *non* PLM parameters, applying discriminative fine-tuning (Kondratyuk and Straka, 2019). We froze PLM parameters for the first few epochs to improve training stability. Adam optimizer (Kingma and Ba, 2015) was used as an optimizer. We applied a 10% and 5% linear learning rate warm-up for all epochs for SI and TC, respectively.

Submitted System: We employed only the official training dataset to train our model. For the SI submission, we generated 24 hyperparameter sets for each PLM, and the top 3 sets chosen on the basis of the validation score for each PLM were used for the stacked generalization. For the TC submission, the top 11 sets amongst 60 hyperparameter sets were used.

Fixed hyperparameter values are shown in Table 1. The tunable hyperparameter set included learning rates and the dropout ratio for the FFNs and BiLSTMs. Optuna (Akiba et al., 2019) was used to generate hyperparameter sets.

The hyperparameter search results for the optimal learning rates are shown in Table 2. Generally, learning rates of non-PLM parameters are larger than those of PLM parameters. Interestingly, most of the learning rates for TC were lower than for SI. This insight suggests that complicated models such as PLMs with BiLSTMs require a larger learning rate to produce a better model.

Metrics: Overlap-based F1 scores for SI and micro-averaged F1 scores for TC were employed in the shared task. Refer to (Da San Martino et al., 2020) for more details.

[12] https://pypi.org/project/pytorch-crf/

hyperparameter	SI	TC
cross-validation folds (k)	5	6
PoS embedding dim	50	50
NE embedding dim	50	50
BiLSTM dim, layers	600, 2	-
`PLM` layer dropout	0.1	0.1
FFN$^{(\text{si\_bio})}$ dim	200	-
FFN$^{(\text{si\_tech})}$ dim	200	-
FFN$^{(\text{si\_sent})}$ dim	500	-
FFN$^{(\text{tc\_bos})}$ dim	-	200
FFN$^{(\text{tc})}$ dim	-	500
FFN$_{\ell}^{(\text{tc})}$ dim	-	500
FFN activation	ReLU	ReLU
λ	0.5	-
Adam β_1, β_2	0.9, 0.999	0.9, 0.999
gradient clipping	5.0	5.0
epochs	18	30
`PLM` frozen first epochs	2	1
batch size	6	5

Table 1: Hyperparameter values

model with `PLM`	`PLM` parameters (SI / TC)	non `PLM` parameters (SI / TC)
BERT	**3.7e-6** / 1.0e-6	**3.5e-4** / 1.6e-4
GPT-2	**1.8e-5** / 1.2e-5	**5.4e-4** / 5.0e-5
RoBERTa	**2.9e-6** / 1.9e-6	**2.8e-4** / 1.0e-4
XLM-RoBERTa	**4.4e-6** / 1.7e-6	**7.3e-4** / 7.0e-5
XLNet	**4.1e-6** / 2.7e-6	**7.3e-4** / 1.5e-4
XLM	3.0e-6 / **6.9e-6**	8.9e-5 / **1.5e-4**

Table 2: Suggested learning rates

team	SI (ranking)	TC (ranking)
Hitachi (ours)	**51.551** (1)	61.732 (3)
ApplicaAI	49.153 (2)	**62.067** (1)
aschern	49.100 (3)	62.011 (2)
LTIatCMU	47.663 (4)	-
UPB	46.060 (5)	54.302 (19)
Solomon	40.683 (15)	58.939 (4)
newsSweeper	42.209 (13)	58.436 (5)

Table 3: Official F scores for test set. We show only top 5 teams for each subtask. **Bold** and underline scores show first and second ranked results, respectively.

model	SI	TC
ensemble all	**49.386**	64.628
ensemble only w/ BERT	46.734	59.454
ensemble only w/ GPT-2	44.161	57.761
ensemble only w/ RoBERTa	49.326	62.841
ensemble only w/ XLM-RoBERTa	47.371	61.524
ensemble only w/ XLNet	47.876	63.406
ensemble only w/ XLM	44.688	60.960
ensemble w/o BERT	48.583	64.440
ensemble w/o GPT-2	48.920	63.688
ensemble w/o RoBERTa	48.065	64.346
ensemble w/o XLM-RoBERTa	49.028	63.688
ensemble w/o XLNet	48.838	63.782
ensemble w/o XLM	49.095	**64.911**

Table 4: Ablation study on development set

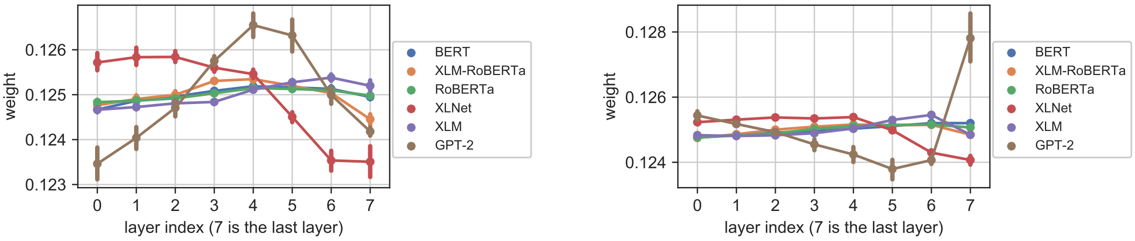
5.1 Results

Table 3 shows the official test results of the top-performing teams for SI and TC. Our proposed SI model outperformed all the other teams with an improvement of more than 2 points. Our team was ranked third for TC; however, the performance of the top three teams seemed to be almost the same.

On the Role of PLMs: To show the role of `PLMs`, we show the ablation results for each `PLM` in Table 4. In the table, we show the performance for the development data. The table shows that RoBERTa and XLNet were generally the best performing models. Given that RoBERTa is a carefully tuned model based on BERT, this result is reasonable. Interestingly, the table also shows that using all `PLMs` in an ensemble was better in most cases. For example, while the GPT-2-based model itself is not a better model for both SI and TC, excluding the GPT-2-based model results in worse performance. This result suggests that stacked generalization was effectively applied.

PLM Layer Weight: Comprehensive overviews of fine-tuned `PLM` states are shown in Figure 3, visualizing weights of `PLM` layers (Clark et al., 2019). The figure illustrates that the last several layers were generally weighted. GPT-2 and XLNet interestingly show different distributions, ranging widely.

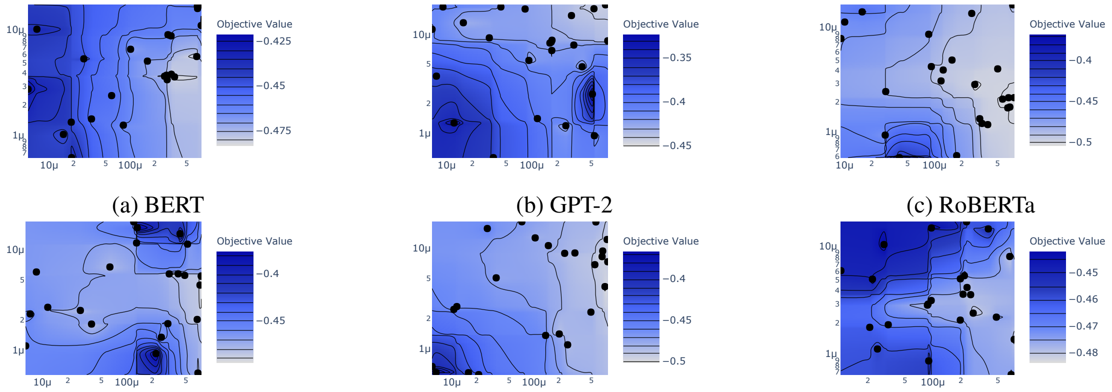
On the Importance of Learning Rate Tuning: Through hyperparameter tuning, we found that tuning a `PLM` learning rate is essential to elicit better results. We show Figure 4 and Figure 5, visualizing the learning rate space for the two parameter groups, that is, `PLM` parameters and non-`PLM` parameters. We found that the SI models required tuning for either group, while the TC models required the tuning of `PLM` parameters rather than non-`PLM` parameters. We attribute this to the complexity of SI models. SI models employ BiLSTM-CRF in `PLMs`, and BiLSTMs are complicated when compared with FFNs in TC. Therefore, SI training requires a higher learning rate for either group. TC models use only FFNs in `PLMs` and therefore require a lower learning rate for non-`PLM` parameters.



(a) Token-level tasks in SI

(b) TC

Figure 3: Attention weight visualization for last eight layers in PLMs



(a) BERT

(b) GPT-2

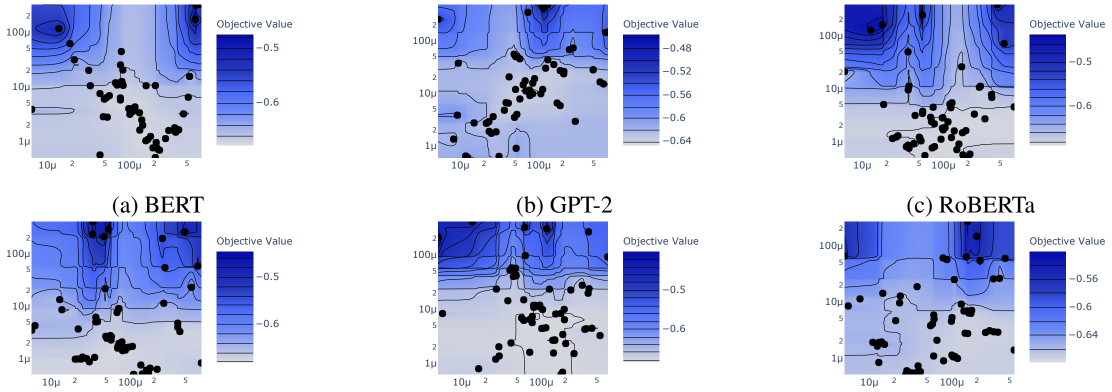
(c) RoBERTa

(d) XLM-RoBERTa

(e) XLNet

(f) XLM

Figure 4: Negative SI validation scores in learning rate space, where X-axis shows learning rate for non-PLM parameters, and Y-axis shows PLM parameters. Each point indicates searched hyperparameter. Note that *brighter colors indicate better performance*.



(a) BERT

(b) GPT-2

(c) RoBERTa

(d) XLM-RoBERTa

(e) XLNet

(f) XLM

Figure 5: Negative TC validation scores in learning rate space

6 Conclusion

We detected propaganda by leveraging heterogeneous pre-trained language models. The results suggested that employing heterogeneous pre-trained language models could result in better performance. Future work includes examining more effective methods for utilizing heterogeneous pre-trained language models.

Acknowledgments

Computational resource of AI Bridging Cloud Infrastructure (ABCI) provided by National Institute of Advanced Industrial Science and Technology (AIST) was used. We thank the anonymous reviewers who gave us insightful comments. We also thank Dr. Masaaki Shimizu for the convenience afforded by these computational resources.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 2623–2631, New York, NY, USA. ACM.

Hani Al-Omari, Malak Abdullah, Ola AlTiti, and Samira Shaikh. 2019. JUSTDeep at NLP4IF 2019 task 1: Propaganda detection using ensemble deep learning models. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 113–118, Hong Kong, China, November. Association for Computational Linguistics.

Alberto Barrón-Cedeño, Israa Jaradat, Giovanni [Da San Martino], and Preslav Nakov. 2019. Proppy: Organizing the news based on their propagandistic content. *Information Processing Management*, 56(5):1849 – 1864.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy, August. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news articles. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*, EMNLP-IJCNLP 2019, Hong Kong, China, November.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, SemEval 2020, Barcelona, Spain, September.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Ali Fadel, Ibraheem Tuffaha, and Mahmoud Al-Ayyoub. 2019. Pretrained ensemble learning for fine-grained propaganda detection. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 139–142, Hong Kong, China, November. Association for Computational Linguistics.

Alex. Graves, Abdel rahman. Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649.

Arthur E. Hoerl and Robert W. Kennard. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Yiqing Hua. 2019. Understanding BERT performance in propaganda analysis. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 135–138, Hong Kong, China, November. Association for Computational Linguistics.

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

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015*.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing universal dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China, November. Association for Computational Linguistics.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*.

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

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

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, June. Association for Computational Linguistics.

Alec Radford and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. In *arXiv*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Claudia Schulz, Steffen Eger, Johannes Daxenberger, Tobias Kahse, and Iryna Gurevych. 2018. Multi-task learning for argumentation mining in low-resource settings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 35–41, New Orleans, Louisiana, June. Association for Computational Linguistics.

Harish Tayyar Madabushi, Elena Kochkina, and Michael Castelle. 2019. Cost-sensitive BERT for generalisable sentence classification on imbalanced data. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 125–134, Hong Kong, China, November. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

George-Alexandru Vlad, Mircea-Adrian Tanase, Cristian Onose, and Dumitru-Clementin Cercel. 2019. Sentence-level propaganda detection in news articles with transfer learning and BERT-BiLSTM-capsule model. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 148–154, Hong Kong, China, November. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

David H. Wolpert. 1992. Stacked generalization. *Neural Networks*, 5:241–259.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5753–5763. Curran Associates, Inc.

Shehel Yoosuf and Yin Yang. 2019. Fine-grained propaganda detection with fine-tuned BERT. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 87–91, Hong Kong, China, November. Association for Computational Linguistics.