Semi-Supervised Learning on Meta Structure: Multi-Task Tagging and Parsing in Low-Resource Scenarios

KyungTae Lim,[1]* Jay Yoon Lee,[2]* Jaime Carbonell,[2] Thierry Poibeau[1]
[1]LATTICE (CNRS & École normale supérieure / PSL & Université Sorbonne nouvelle Paris 3 / USPC), Paris
[2]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA
[1]{kyungtae.lim, thierry.poibeau}@ens.fr
[2]{jaylee, jgc}@cs.cmu.edu

Abstract

Multi-view learning makes use of diverse models arising from multiple sources of input or different feature subsets for the same task. For example, a given natural language processing task can combine evidence from models arising from character, morpheme, lexical, or phrasal views. The most common strategy with multi-view learning, especially popular in the neural network community, is to unify multiple representations into one unified vector through concatenation, averaging, or pooling, and then build a single-view model on top of the unified representation. As an alternative, we examine whether building one model per view and then unifying the different models can lead to improvements, especially in low-resource scenarios. More specifically, taking inspiration from co-training methods, we propose a semi-supervised learning approach based on multi-view models through consensus promotion, and investigate whether this improves overall performance. To test the multi-view hypothesis, we use moderately low-resource scenarios for nine languages and test the performance of the joint model for part-of-speech tagging and dependency parsing. The proposed model shows significant improvements across the test cases, with average gains of $-0.9 \sim +9.3$ labeled attachment score (LAS) points. We also investigate the effect of unlabeled data on the proposed model by varying the amount of training data and by using different domains of unlabeled data.

1 Introduction

Multi-view data consist of different manifestations of the same data, often in the form of different features, and such data are abundant in real-world applications (Xu, Tao, and Xu 2013). Character-, word- level representations, stem, prefix, and suffix are examples of multi-view data in Natural Language Processing (NLP).

The use of multi-view data has resulted in considerable success in various NLP problems. Combining different word representations at the character, token, or sub-word levels has proven to be helpful for dependency parsing (Botha et al. 2017; Andor et al. 2016), Part-of-Speech (POS) tagging (Plank, Søgaard, and Goldberg 2016), and other NLP tasks.

Given multiple views, a simple but popular approach is to unify multiple representations into a combined one through concatenation, averaging, or pooling. This approach is especially popular in neural networks as it is very straightforward to concatenate multiple representations without any modification of the model. All the aforementioned work also considered this approach. However, is it the best usage of multi-view data? A simple input concatenation can lead to overfitting problem as the model might ignore the specific statistical property of each view (Zhao et al. 2017).

Recently, META-BILSTM (Bohnet et al. 2018) was proposed to extend the naive solution of concatenating input representations in the context of POS tagging, and it showed superior performance compared to simple view concatenation on input representations. META-BILSTM builds a single-view model of each view (lower layer) and concatenates the series of single-view-model outputs to form an input to the meta layer, as shown in Figure 2. All the components of META-BILSTM (per-view models and meta layer) are trained jointly, as expressed in Eq.(2).

In this study, we first examine whether META-BILSTM can be beneficial in the context of more complex tasks, namely multi-tasking in POS tagging and dependency parsing. The study then proposes `Co-meta`, a semi-supervised approach, to improve each single-view model through the consensus promotion of the multiple single-view models on unlabeled data. The proposed `Co-meta` is motivated by Co-Training (Blum and Mitchell 1998), a classic approach similar to multi-view learning, which enables exploration of unlabeled data and is known to be helpful in low-resource settings. Overall, Co-Training and many of its variants improve the multi-view models by maximizing agreement between the multi-view models on unlabeled data, and thus can improve performance in low-resource settings.

Thus, this study raises the question of whether classical Co-Training style approaches can further improve the META-BILSTM model in low-resource settings. Specifically, we explore two questions: (1) can respective models from different views learn from each other on unlabeled data? Moreover, (2) can this help the performance of low-resource models? We study whether improving each multi-view model by promoting the consensus in a Semi-Supervised Learning

[*] Equal contribution.

(SSL) fashion can lead to learning better meta models in the context of joint tagging and dependency parsing.

Once we apply META-BiLSTM, we obtain several parsing models trained by each view. Then the main challenge that arises with regard to our SSL approach (Co-meta) is deciding *what* and *how much* a single view should learn from other views. We suggest three different methods for determining *what* to learn from each other, namely, Entropy, Voting, and the Ensemble-based approach. Then, to determine *how much* to learn from the determined example, we introduce confidence score in section 3.3.

We employ our SSL methods and META-BiLSTM on top of the graph-based parser with a bi-affine classifier proposed by (Dozat, Qi, and Manning 2017), and investigate the effectiveness of our approach on both low- and high-resource scenario experiment setups using the Universal Dependency 2.3 dataset (Zeman and others 2018). Co-meta, the proposed model shows consistent improvement across the test cases, with an average of $-0.9 \sim +9.3$ Labeled Attachment Score (LAS) gains in low-resource and $0.2 \sim 1.1$ in high-resource settings, respectively. The study also investigates whether the proposed method depends on unlabeled data by changing the amount and varying the domains of unlabeled data, and its effect on the proposed model. In summary, our contributions to joint parsing are as follows:

1. Proposal of a new formulation Co-meta that leverages consensus promotion on top of a META-BiLSTM model.
2. Analysis of the relation of each multi-view model performance to that of the meta model.
3. Exploring different semi-supervised scenarios, where the amount of unlabeled data and the domains of unlabeled data are varying.
4. Generalization of META-BiLSTM and Co-meta by expanding an additional-view model on top of the existing model using external word embedding.

2 Related Work

2.1 Dependency Parsing with Multi-Task Structure

Dependency parsing is an essential component of many NLP applications because of its ability to capture complex relational information in a sentence. Typically, the goal of dependency parsing is to derive a tree structure for a sentence x = (w_1, w_2 ...w_n) following a given dependency grammar. A syntactic dependency tree consists of dependency arcs (each arc is a relation between a $Head$, w_h, and one or more dependent words w_m); each arc is labeled Dep to define the relation between w_m and w_h. Dependency parsing is widely used for tasks such as named entity recognition (Kazama and Torisawa 2008), discourse understanding (Sagae 2009), and information extraction (Fares et al. 2018).

Recent breakthroughs in multi-task learning have made it possible to effectively perform different tasks with the same model. The multi-task approach enriches context-sensitive feature representations by learning different tasks using shared parameters (Hashimoto et al. 2016). In NLP, this approach has been widely used to learn joint models performing tagging and parsing simultaneously, and all state-of-the-art (SOTA) models now use a multi-task structure. In general, given an input sentence x and a set of gold labels $y = (l_1$, $l_2 ... l_n)$, where each l_i consists of labels for tagging and parsing, the goal of the multi-task structure is to train a joint model that can provide at the simultaneously a POS tagger and a dependency parser.

There are many variants of multi-task learning for tagging and parsing. These variants consist in models sharing parameters between the tasks (Straka 2018) and models sharing variants (Che et al. 2018; Lim et al. 2018). On top of this, recent systems trained with Language Model (LM) representations have shown even better results. One of these models, ELMo (Peters et al. 2018), which is trained with unsupervised textual representations using BiLSTM. Models with ELMo obtained the best performance in the 2018 CoNLL shared task (Che et al. 2018; Lim et al. 2018). Another more-recent and cutting-edge LM, BERT (Devlin et al. 2019), which is trained by bidirectional transformers with a masked language model strategy, shows outstanding results in parsing (Kondratyuk 2019; Kulmizev et al. 2019). While many variants exist, all these models basically produce a single parser and tagger based on a single concatenated view. In contrast, Bohnet et al. proposed an approach to build several POS taggers trained by individual lexical representations and generated a multi-view model only for POS tagging.

2.2 Co-Training

The standard multi-view learning approaches try to learn a model by jointly optimizing all the multi-view models arising from different views as opposed to combining input level multi-view data. The most representative and one of the earliest multi-view learning methods is Co-Training (Blum and Mitchell, 1998). Co-Training and many variants of Co-Training many of its variants (Nigam and Ghani 2000; Muslea, Minton, and Knoblock 2002; Yu et al. 2011) try to maximize the mutual agreement of multi-view models on unlabeled data promoting *consensus* principle. The unified model is said to have improved when each view provides some knowledge that the other views do not possess; that is, different views hold *complementary* information.

3 Proposed Approach

We first consider the baseline model introduced by (Lim et al. 2018) and extend it so as to get a multi-view model structure following (Bohnet et al. 2018). Then, in section 3.3, we propose a new semi-supervised learning (SSL) approach called Co-meta. We detail the proposed loss function (Section 3.4) for Co-Training while taking into account the provided meta structure.

3.1 The BASELINE Model

As it is known that using information from multiple views yield better performance, most SOTA multi-task parsers use both word-level and character-level views to get a lexical embedding $v_{1:n}^{(wc)}$ from a sequence of n words $w_{1:n}$. Most of these approaches simply concatenate word embedding $v_i^{(w)}$

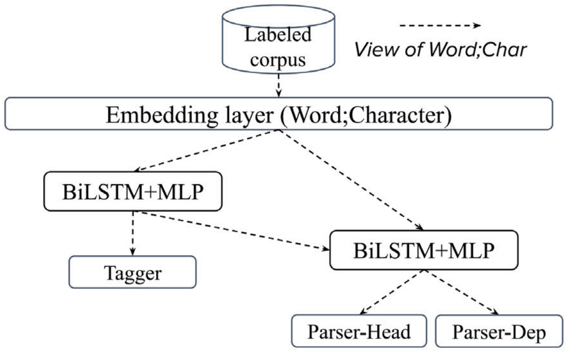

Figure 1: Overall structure of our baseline model.

and the character-level embedding $v_i^{(c)}$ of w_i to form $v_i^{(wc)}$. For example, Figure 1 shows a multi-task parsing architecture for low-resource scenarios proposed by (Lim et al. 2018). It obtained good results on the CoNLL 2018 shared task (Zeman et al. 2018). Specifically, the parser transforms the sequence of shared lexical representation $v_i^{(wc)}$ to a context-sensitive vector contextualized by BiLSTM with a hidden layer r_0 as:

$$h_i^{(pos)} = BiLSTM(r_0^{(pos)}, (v_1^{(wc)}, .., v_n^{(wc)}))_i$$
$$h_i^{(dep)} = BiLSTM(r_0^{(dep)}, (v_1^{(wc)}, .., v_n^{(wc)}))_i$$

The system uses vector $h_i^{(pos)}$ to predict POS with a Multi-layer Perceptron (MLP) classifier, and $h_i^{(dep)}$ for $Head$ and Dep with a bi-affine classifier (Dozat and Manning 2016). During training, it learns the parameters of the network θ that maximize the probability $P(y_j|x_j, \theta)$ from the training set T based on the conditional negative log-likelihood loss of our baseline B_loss(θ). Thus,

$$\text{B\_loss} = \sum_{(x_j,y_j)\in T} -\log P(y_j|x_j, \theta) \quad (1)$$
$$\hat{y} = \arg\max_y P(y|x_j, \theta)$$

where $(x_j, y_j) \in T$ denotes an element from the training set T, y is a set of gold labels (l^{POS}, l^{Head}, l^{Dep}), and $\hat{y}$ is a set of predicted labels. The model of Lim et al.[1] is subsequently used as the BASELINE model.

3.2 Supervised Learning on META(META-BASE)

In order to examine whether a multi-view learning approach similar to that of Bohnet et al. would also be helpful to perform tagging and parsing jointly, we propose the meta structure shown in Figure 2. We use Lim et al.'s multi-task structure of tagging and parsing as our default single-view model and call the overall system META-BASE.

We define a model M^{vi} for each view $vi \in V$, where V is the set of all views. For example, Figure 2 contains different views for word, character, and meta levels, and V is expressed as $V = \{\text{word,char,meta}\}$. Each model M^{vi} consists of a BiLSTM^{vi} that contextualizes its view with a representation h_i^{vi} for word w_i, and an MLP classifier to predict POS tag and a bi-affine classifier (Dozat and Manning 2016) to predict parsing outputs $Head$ and Dep. As the input of each view, M^{word} and M^{char} consume the word- and character-level embedding, respectively, and M^{meta} consumes the concatenation of two models' contextualized outputs as $[h_i^{word};h_i^{char}]$. Each M^{vi} is parameterized by the network parameter θ^{vi}, and the overall network parameter θ is defined as the union of the network parameters of all views, that is, $\theta = \cup_{vi\in V}\theta^{vi}$.

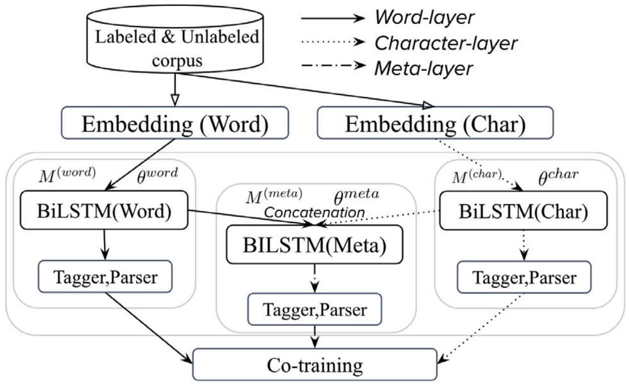

Figure 2: Overall structure of our Co-meta model.

During supervised learning, we train θ to maximize the probability $P(y_j|x_j, \theta)$ for the input and labeled instance pair (x_j, y_j) in the training set T by optimizing over the supervised loss (S_loss) as follows:

$$\text{S\_loss} = \sum_{(x_j,y_j)\in T} -\log P(y_j|x_j, \theta) \quad (2)$$

which is simply the standard cross entropy loss, where $\log P(y_j|x_j, \theta)$ stands for $\sum_{vi\in V} \log P(y_j|x_j, \theta^{vi})$ for brevity. Note that the predicted POS results are added to the parser's classifier as an embedding (learnable parameters) during training.

3.3 Co-meta

Co-meta stands for the Co-Training approach on the meta structure. The main idea of Co-Training is to augment training data with each model's confident prediction on unlabeled data so that each model can learn from other models' predictions. While not exactly following the Co-Train approach, we adopt the idea of one model teaching other models. We propose to extract the best possible parsing result using all models' predictions as $\hat{y}^*$ on a given instance x in unlabeled set U, and make each single-view model learn from $\hat{y}^*$ by optimizing over the proposed unsupervised loss (C_loss) as follows:

$$\text{C\_loss} = -\sum_{vi\in V\setminus\{\text{meta}\}} \sum_{x\in U} g(\hat{y}^*, \hat{y}^{vi}) \log P(\hat{y}^*|x, \theta^{vi}). \quad (3)$$

[1] The parser achieved the 2nd and 4th ranks with regard to UAS and LAS, respectively, out of 27 teams in the CoNLL 2018 shared task.

Here, $\hat{y}^{vi} = \arg\max_y P(y|x, \theta^{vi})$ stands for the output for view vi and the $g(\hat{y}^*, \hat{y}^{vi})$ stands for the confidence score, which measures the confidence of $\hat{y}^{vi}$ with respect to $\hat{y}^*$. The ways to obtain $\hat{y}^*$ can be divided into three variants depending on how one extracts $\hat{y}^*$. We detail the notions of Entropy-based, Voting-based, and Ensemble-based extraction.

- **Entropy-based extraction** selects the entire prediction of model M^{vi^*} and set $\hat{y}^* = \hat{y}^{vi^*}$ where the prediction of view vi^* has the lowest entropy for its prediction score , i.e. $vi^* = \text{argmax}_{vi \in V} P(\hat{y}^{vi}|x, \theta^{vi})$. In the entropy-based approach, the view vi^* *only teaches other views and does not teach itself.*
- **Voting-based extraction** selects the most popular label among the three models for each word w_m. When there is no agreement between the output of each model, we select the prediction of $M^{(meta)}$.
- **Ensemble-based extraction** selects $\hat{y}^*$ using an ensemble method, that is, $\hat{y}^* = softmax(\sum_{vi \in V} P(\hat{y}^{vi}|x, \theta^{vi}))$.

In addition, we scale the loss function with the confidence score $g(\hat{y}^*, \hat{y}^{vi})$, which measures the similarity between the two arguments. The idea is to assess how much confidence one should have in updating model θ^{vi} with instance $\hat{y}^*$. We hypothesize that if the prediction $\hat{y}^{vi}$ has a similar structure to the extracted $\hat{y}^*$, then the vi-view model is aligned with the extracted output and thus can confidently learn from $\hat{y}^*$. In more detail, confidence score $g(\hat{y}^{vi}, \hat{y}^{vj}) = \sum_{t=1}^{n} I({y^{vi}}_t, {y^{vj}}_t)/n$ ranging from 0 to 1 is a simple agreement measure between $\hat{y}^{vi}, \hat{y}^{vj}$ normalized by the sentence length n. Note that we update the parameters of each view model but do not update the parameters of $M^{(meta)}$ using C_loss to avoid overfitting.

The idea of learning from model's own prediction was explored by (Dong and Schäfer 2011) in the context of self-training but without the confidence score. In our experiments, all the models without a confidence score showed a decrease of performance for all the three variants of $\hat{y}^*$.

3.4 Joint Semi-Supervised Learning

While labeled data T is small in low-resource scenarios, we often have larger unlabeled data U. We thus need to leverage the supervised model Eq.(2) using unlabeled data. Since our C_loss only requires prediction result $\hat{y}$, we can train both T and U as a joint loss (J_loss) as follows:

$$\begin{aligned} \text{J\_loss} = &\sum_{(x_j, y_j) \in T} -\log P(y_j|x_j, \theta) \\ - &\sum_{vi \in V} \sum_{x_k \in U} g(\hat{y}^*_k, \hat{y}^{vi}_k) \log P(\hat{y}^*_k|x_k, \theta^{vi}) \end{aligned} \tag{4}$$

where $T \subseteq U$ might apply to U, T when using T without labels. During the joint learning phase, we use the individual $CrossEntropy$ objective function to compute all the losses with an $Adam$ optimizer. In what follows, let's call `Co-meta` the training process with J_loss on the meta-LSTM structure.

4 Experiments

4.1 Data Sets

We evaluate `Co-meta` on the Universal Dependency 2.3[2] test set for nine languages, following the criteria from (de Lhoneux, Stymne, and Nivre 2017), with regard to typological variety, geographical distance, and the quality of the treebanks. Our testing languages are thus Ancient Greek, Chinese, Czech, English, Finnish, Greek, Hebrew, Kazakh, and Tamil. During training, we use pre-trained word embeddings[3] and unlabeled data[4] from the CoNLL 2018 shared task to initialize our word embedding $v^{(w)}$ and the SSL presented in the previous section. When we employ Language Models, we use pretrained models provided by (Lim et al. 2018) for ELMo and Google[5] for BERT. We use the gold segmentation result for the training and test data.

4.2 Evaluation Metrics

There are two major evaluation metrics in dependency parsing. The Unlabeled Attachment Score (UAS) is used to evaluate the structure of a dependency graph. It measures to what extent the structure of the parsed tree is correct, without taking into account the labels on the different arcs of the tree. The Labeled Attachment Score (LAS) is the same as UAS, but takes into account dependency labels.

As for POS tagging, we measure the percentage of words that are assigned the correct POS label. We evaluate our tagger and parser based on the official evaluation metric provided by the CoNLL 2018 shared task[6].

4.3 Experimental Setup

We sample 50 instances from labeled data as a training set to test the low-resource scenario following (Guo et al. 2016). In addition, we test our models on extremely low-resource scenarios to investigate the effect of our semi-supervised approach. We borrow hyperparameter settings from the BASELINE and apply them on the single-view layers in the META-BiLSTM structure. In each epoch, we run over the training data with a batch size of 2 and run only a batch from randomly chosen unsupervised data. We evaluate our models on the test sets, and report the average of the three best performing results, trained with different initial seeds, within 1,000 epochs. All the reported scores without any mention are based on the scores from the meta-layer output.

5 Results

Our study has different goals: (1) study the impact of multi-view based learning, META-BASE `Co-meta`, on tagging and parsing in low-resource scenarios, (2) check whether `Co-meta` can increase the consensus between single-view models and the effect of this promoted consensus on the performance of each-view model and on the overall

[2] http://hdl.handle.net/11234/1-2895

[3] http://hdl.handle.net/11234/1-1989

[4] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-1989#

[5] https://github.com/google-research/bert

[6] https://universaldependencies.org/conll18/evaluation.html

Table 1: LAS and UPOS scores of $M^{(meta)}$ model output on the test set using 50 training sentences and unlabeled sentences based using Co-meta, META-BASE, and our BASELINE model (Lim et al. 2018). We report META-BASE to decompose the performance gain into the gains due to META-BASE (supervised) and Co-meta (SSL). *Kazakh only has 31 labeled instances. Thus we use only 31 sentences and its unlabeled data are sourced from Wikipedia whereas other languages take the unlabeled data from the given training corpus after removing label information.

		VOTING		ENTROPY		ENSEMBLE		META-BASE		BASELINE	
corpus	unlabeled	LAS	POS	LAS	POS	LAS	POS	LAS	POS	LAS	POS
cs_cac (Czech)	23478	47.4	79.4	47.4	79.7	**48.7**	**81.4**	45.9	79.0	39.4	74.6
fi_ftb (Finnish)	14981	21.7	43.2	22.0	44.7	21.8	43.5	21.9	**44.6**	**22.6**	39.2
en_ewt (English)	12543	45.1	75.7	46.3	**76.7**	**46.5**	76.3	45.4	75.2	42.8	71.1
grc_perseus (Ancient Greek)	11460	30.8	70.1	**31.7**	**70.9**	31.3	70.7	30.9	70.4	29.5	65.8
he_htb (Hebrew)	5240	47.9	76.9	47.8	77.2	**48.4**	**77.4**	47.6	76.7	45.1	75.2
zh_gsd (Chinese)	3997	36.1	70.7	35.1	70.8	**36.9**	**71.1**	35.1	70.6	34.8	68.7
el_bdt (Greek)	1162	60.0	84.3	**60.6**	83.2	60.5	**84.2**	57.8	82.6	51.7	80.0
ta_ttb (Tamil)	400	38.1	69.1	39.0	**69.7**	**40.0**	69.3	38.3	67.3	34.0	61.9
kk_ktb (Kazakh)*	12000*	27.6	56.9	27.9	57.0	**28.7**	57.1	27.8	**57.7**	26.2	53.0
Average	-	39.4	69.6	39.8	70.0	**40.3**	**70.1**	39.0	69.3	36.2	65.5

Table 2: LAS on Greek(el_bdt) corpus for each model, with the average confidence score $g(\hat{y})$ comparing $M^{(word)}$ and $M^{(char)}$ over the entire test set using 100 training sentences.

Method	WORD	CHAR	META	CONFIDENCE
ENTROPY	**61.8**	66.7	**69.1**	0.871
ENSEMBLE	61.4	**66.9**	69.0	**0.879**
WITHOUT	57.6	65.2	67.4	0.799

META-BILSTM system, (3) study the effect of unlabeled data on Co-meta, and finally (4) investigate to what extent the efficacy of Co-meta remains when the approach is applied to high-resource scenarios.

5.1 Results in Low-Resource Settings

Impact of Multi-View Learning. Table 1 shows the experimental results of $M^{(meta)}$ on the test data of each language. We see that the proposed Co-Training method shows average performance gains of $-0.9 \sim +9.3$ LAS points in parsing and $+1.7 \sim 6.9$ points in tagging compared to BASELINE.

Note that the proposed META-BASE approach also shows a LAS improvement of $-0.6 \sim +6.5$ for BASELINE as well. Breaking down the contribution of improvement, Co-meta shows $-0.3 \sim +2.8$ LAS improvement over META-BASE and this improvement is comparable to the improvement of META-BASE over BASELINE.

Comparison of Co-meta Variants. When we compare the three proposed Co-Training approaches, one can see that the ENSEMBLE approach seems to work better than ENTROPY, and VOTING is always worst. This is because the best-voted labels for each token do not guarantee to get an optimal structure over the parse tree at the sentence-level, since the VOTING model has a relatively high chance of learning from the inconsistent graph that has multi-roots and cycling heads among tokens.

Lastly, we also try running Co-meta experiments without confidence scores, i.e., we set the confidence score as 1. We find that, with this configuration, performance always decreases in comparison to META-BASE, and thus, we conclude that the proposed confidence score plays a major role in stabilizing the Co-Train approach.

Interaction among the layers? More detailed per-layer analysis of the LAS scores is available in Table 2 for the case of Greek corpus. Among the three views, CHAR always outperforms WORD, and all three views improve after using Co-meta: improvements of 1.6-1.7 LAS point for META, 1.5-1.7 for CHAR and 3.8-4.2 for WORD.

We make three interesting observations. First, we note that the model with lower performance, namely WORD view in our example, always benefits the most from other better-performing views. Second, the evolution of low-performing views towards better results has a positive effect on META view, and thus on the overall performance. While the score CHAR increases by 1.5, META increases by 1.7. If the lower-performing-view model was not helping, then the improvement would be upper-bounded by the performance gain of the higher-performing model. Note that we do not update the META layer $\theta^{(meta)}$ when using Co-meta, and all the gains result from the improvements of the single-view layers. Lastly, we can observe the CONFIDENCE scores between *word* and *char* views on the last column increase when we apply Co-meta. As the Higher CONFIDENCE denotes that models predict a similar tree structure, we can confirm that Co-meta indeed promotes the consensus between the views.

Sensitivity to the Domain of the Unlabeled Set. In Table 3, we investigated a more realistic scenario for our semi-supervised approach for two languages, Chinese and Greek, by using out-of-domain data: Wikipedia and a crawled corpus. In the case of Chinese, the crawled out-domain corpus shows

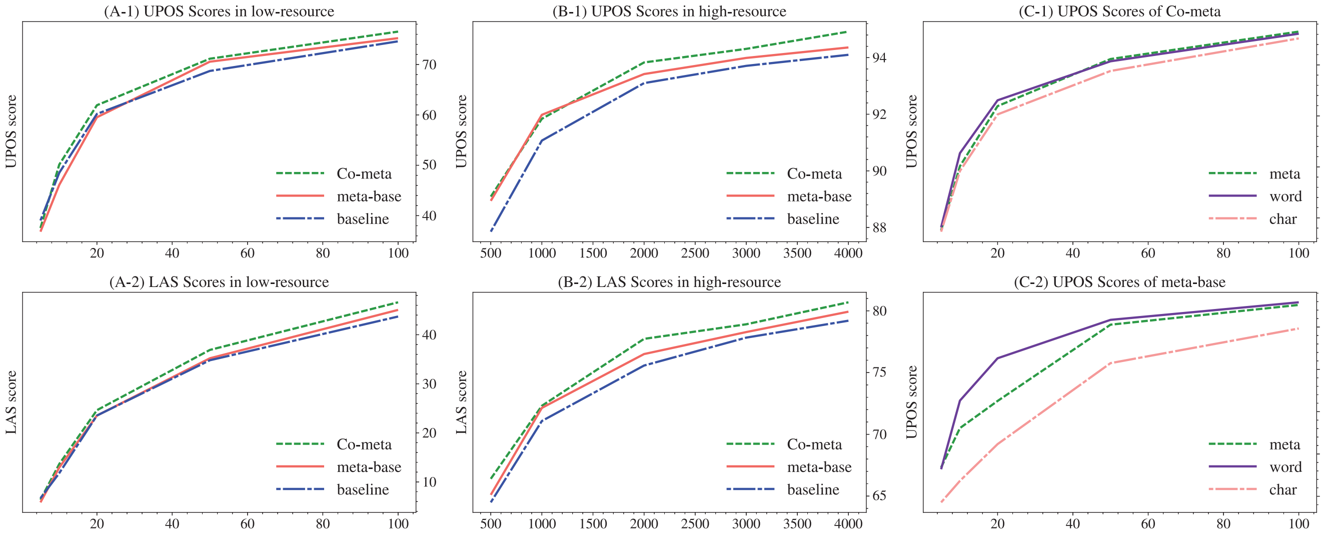

Figure 3: Evaluation results for Chinese (zh_gsd) based on the different sizes of the train set and proposed models. We apply ENSEMBLE based Co−meta with the fixed size of 12k unlabeled sentences while varying training set size.

Table 3: Scores of Co−meta with the ENSEMBLE method on different domains of unlabeled data with 100 training sentences.

Labeled	Unlabeled	size	LAS	UAS	POS
el_bdt (Greek)	el_bdt	1162	**69.0**	**75.6**	88.5
	wikipedia	12000	68.7	75.1	**88.7**
	crawl	12000	68.3	74.8	88.4
zh_gsd (Chinese)	zh_gsd	3997	45.3	57.9	76.9
	wikipedia	12000	**46.3**	**59.1**	77.6
	crawl	12000	46.1	59.0	**77.8**

better results than the in-domain corpus for both ENTROPY-based and ENSEMBLE-based Co−meta, by up to 1.1 UAS and 0.9 POS points. In contrast, for Greek, the in-domain corpus (el_bdt) shows a better result than the out-domain corpus even when the size of el_bdt is only about 13% of the others. We conjecture that as the Chinese has large character sets, the exposure to diverse characters helps learning regardless of the domain.

Effect of Training Size on Performance. Table 1 shows positive results for the Co−meta given fixed size train data. However, would Co−meta be useful even with extremely low resource scenarios (¡50 sentences)? And also in a more favorable scenario, when more resources are available for training (e.g. ¿1000 sentences)? To answer these questions, we conducted an experiment using the zh_gsd (Chinese) corpus with training sets of different sizes, but with a fixed set of 12k unlabeled data. The results are visible in Figures 3(A,B).

Figure 3(A) shows our results for the lower resource scenario (with less than 50 sentences for training). Co−meta outperforms META-BASE and BASELINE except when only five sentences are used for training. We conjecture that this result is attributable to the fact that too little vocabulary is used to allow meaningful generalization. A similar behavior was observed for fi_ftb in Table 1: in this experiment, there is only 241 token available for fi_ftb, whereas other languages had on average $\sim$1388. However, as observed in Figure 3, once we expand the labeled instances (¿20 sentences), Co−meta and META-BASE always outperform BASELINE, both in lower (3A) and higher resource (3B) settings. Also note that Co−meta always outperforms META-BASE, including when one only has 5 labeled instances for training.

We can refine our analysis by examining the different layers of META-BASE and Co−meta that appear on Figure 3 A-1. META-BASE is detailed on Figure 3 C-2 and Co−meta on Figure 3 C-1. In most cases, META stays close to the highest performing view (the WORD layer for most cases). One interesting fact is that the WORD as well as meta layer of META structures outperform the BASELINE which is built on a combined view.

The biggest contrast between Co−meta and META-BASE is the gap between the performances of the WORD and the CHAR layers. A closer look at META-BASE(C-2) seems to indicate that the performance of the META layer cannot differ too much from the lower-performing layer (CHAR in our case). When the gap between WORD and CHAR becomes too large (¿5 points), then the performance gain of META layer is parallel to that of CHAR layer for train size of 10–50 even when the WORD layer makes steeper performance gains. In contrast, the Co−meta's META layer from 3(C-1) shows more stable performance as the gap between CHAR and WORD is minimal as the two layers learn from each other.

To summarize from Table 1 and Figure 3, the proposed SSL approach is always beneficial for the META-BILSTM structure when comparing the LAS scores between Co−meta and META-BASE. However, the META-BILSTM structure itself might not benefit when too few tokens exist in the train set. In general, we hypothesis that for META-BILSTM structure

Table 4: LAS for the English (en_ewt) corpus for each model, with the external language models with the entire train set.

Model	LM	LAS	UAS	POS
UDPIPE (2019)	-	86.97	89.63	96.29
BASELINE (2018)	-	86.82	89.63	**96.31**
METABASE	-	86.95	89.61	96.19
CO-META	-	**87.01**	**89.68**	96.17
BASELINE (2018)	ELMo	88.14	91.07	96.83
METABASE	ELMo	88.28	91.19	96.90
CO-META	ELMo	88.25	91.19	96.84
UDIFY (2019)	BERT-MULTI	88.50	90.96	96.21
UUPARSER (2019)	BERT-MULTI	87.80	-	-
BASELINE	BERT-MULTI	89.34	91.70	96.66
METABASE	BERT-MULTI	89.49	92.01	96.75
CO-META	BERT-MULTI	89.52	91.99	96.80
CO-META	BERT-BASE	**89.98**	**92.25**	**97.03**

to be useful, the train set should consist of more than 300 tokens (more than 20 sentences) to provide generality.

5.2 Results in High-Resource Settings

Although the lack of annotated resources for many languages has given rise to low-resource approaches, several languages exist with plenty of resources. We thus need to examine whether our approach is also effective in more favorable setting, when large scale resources are available. A comprehensive overview is shown in Table 4, where different systems using no language model (first part of the table), or ELMo (Peters et al. 2018) or BERT (Devlin et al. 2019) language models are evaluated.

Table 4 includes a comparison of our results using the approach presented in this paper with four state-of-the-art systems. The first system is BASELINE (introduced in section 3.1), which obtained the best LAS measure for English in the 2018 CoNLL shared task. The second is UDPIPE (Straka 2018; Kondratyuk 2019) which was one of the best performing systems during the 2018 CoNLL shared task (best MLAS score, that combines tagging and parsing, and 2nd for the average LAS score). UDPIPE uses a multi-task learning approach with a loosely-joint LSTM layer between tagger and parser. The third system is UDIFY (Kondratyuk 2019) (derived from UDPIPE), where the LSTM layer is replaced with BERT embedding, which is in turn fine-tuned during training. The fourth system is UUPARSER wherein concatenated word, character and BERT embedding serves as an input, i.e., h_i =[$v^{(wc)}$;$v^{(bert)}$].

Effect of `Co-meta` On High-Resource Settings without LMs. By expanding the baseline with our meta-LSTM and SSL approach, we observe a slight improvement of up to 0.19 and 0.04 points against the BASELINE and UDPIPE, respectively. In contrast, we find that both META-BASE and `Co-meta` slightly underperform the BASELINE in tagging, which goes against our intuition. One possible reason might be that there is enough data to get accurate results using a supervised learning approach while SSL suffers from unexpected surface sequences. Another evidence of this is that SSL did not bring further improvement when using more than 10,000 training sentences. In contrast, interestingly, Chinese for which we had a relatively small train set (3,997), is positively affected by SSL, with a gain of up to 0.21 LAS points comparing to UDPIPE, 1.12 points with BASELINE. We assume that the main reason for this is the character set. Languages with a bigger character set size and little training data gain more influence with SSL.

Effect of `Co-meta` On High-Resource Settings with LMs. While we train our model with a LM, we concatenate the last layer of the LM embedding with the input of the $BiLSTM^{(meta)}$ presented in the previous section. Finally, the input of our meta model consists of three different contextualized features as [$h_i^{(word)}$;$h_i^{(char)}$;$v_i^{(lm)}$].

On average, adding a LM provides excellent results for both dependency parsing and POS tagging outperforming cases without LMs by large margins, up to 1.27 LAS for ELMo and 2.97 for BERT. Furthermore, our parser with `Co-meta` globally shows better results than the state-of-the-art parsers that use ELMo (Lim et al. 2018) and BERT-Multilingual model (Kondratyuk 2019). However, it should be noted that the UDIFY model used by (Kondratyuk 2019) (that includes Bert-Multilingual as a LM) was first trained with 75 different languages using Universal Dependency corpora and then tuned for English, and it is not clear how this training process affects the performance. Thus, we add the results of UUPARSER and BASELINE with BERT to represent fine-tuning in a monolingual way only and still found that CO-META+BERT-MULTI shows better performance.

We generalized `Co-meta` by adding an additional view: LM embedding. We conclude that `Co-meta` can, surprisingly, result in positive effects by more than 1–1.7 points compared to competing models and by 0.2 compared to the BASELINE even in a high-resource setting.

6 Conclusion

In this paper, we have presented a multi-view learning strategy for joint POS tagging and parsing using Co-Training methods. The proposed entropy and ensemble-based `Co-meta` yield the best result. This strategy is especially well suited for low-resource scenarios, when only a very small sample of annotated data is available, along with larger quantities of unlabeled data. Our experiment shows statistically significant gains (-0.9 to +9.3 points compared to the baseline), largely due to the proper integration of unlabeled data in the learning process. As future research, we wish to apply `Co-meta` to other sequence-labeling tasks such as Named Entity Recognition and semantic role labeling.

7 Acknowledgments

We want to thank the AAAI reviewers for their useful comments. Thierry Poibeau is a member of PrAIRIe (Paris Artificial Intelligence Research Institute) that partially funded this research. KyungTae Lim is supported by the ERA-NET Atlantis project.

References

Andor, D.; Alberti, C.; Weiss, D.; Severyn, A.; Presta, A.; Ganchev, K.; Petrov, S.; and Collins, M. 2016. Globally normalized transition-based neural networks. *CoRR* abs/1603.06042.

Blum, A., and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, 92–100. ACM.

Bohnet, B.; McDonald, R. T.; Simões, G.; Andor, D.; Pitler, E.; and Maynez, J. 2018. Morphosyntactic tagging with a meta-bilstm model over context sensitive token encodings. *CoRR* abs/1805.08237.

Botha, J. A.; Pitler, E.; Ma, J.; Bakalov, A.; Salcianu, A.; Weiss, D.; McDonald, R. T.; and Petrov, S. 2017. Natural language processing with small feed-forward networks. *CoRR* abs/1708.00214.

Che, W.; Liu, Y.; Wang, Y.; Zheng, B.; and Liu, T. 2018. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, 55–64. Brussels, Belgium: ACL.

de Lhoneux, M.; Stymne, S.; and Nivre, J. 2017. Old school vs. new school: Comparing transition-based parsers with and without neural network enhancement. In *In Proceedings of the 15th Treebanks and Linguistic Theories Workshop*, 99–110.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 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)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Dong, C., and Schäfer, U. 2011. Ensemble-style self-training on citation classification. In *Proceedings of 5th international joint conference on natural language processing*, 623–631.

Dozat, T., and Manning, C. D. 2016. Deep biaffine attention for neural dependency parsing. *CoRR* abs/1611.01734.

Dozat, T.; Qi, P.; and Manning, C. D. 2017. Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, 20–30. Vancouver, Canada: ACL.

Fares, M.; Oepen, S.; Øvrelid, L.; Björne, J.; and Johansson, R. 2018. The 2018 shared task on extrinsic parser evaluation: On the downstream utility of English universal dependency parsers. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, 22–33. Brussels, Belgium: ACL.

Guo, J.; Che, W.; Yarowsky, D.; Wang, H.; and Liu, T. 2016. A representation learning framework for multi-source transfer parsing. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Hashimoto, K.; Xiong, C.; Tsuruoka, Y.; and Socher, R. 2016. A joint many-task model: Growing a neural network for multiple NLP tasks. *CoRR* abs/1611.01587.

Kazama, J., and Torisawa, K. 2008. Inducing gazetteers for named entity recognition by large-scale clustering of dependency relations. In *proceedings of ACL-08*, 407–415.

Kondratyuk, D. 2019. 75 languages, 1 model: Parsing universal dependencies universally. *CoRR* abs/1904.02099.

Kulmizev, A.; de Lhoneux, M.; Gontrum, J.; Fano, E.; and Nivre, J. 2019. Deep contextualized word embeddings in transition-based and graph-based dependency parsing–a tale of two parsers revisited. *arXiv preprint arXiv:1908.07397*.

Lim, K.; Park, C.; Lee, C.; and Poibeau, T. 2018. SEx BiST: A multi-source trainable parser with deep contextualized lexical representations. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, 143–152. Brussels, Belgium: ACL.

Muslea, I.; Minton, S.; and Knoblock, C. A. 2002. Active+ semi-supervised learning= robust multi-view learning. In *ICML*, volume 2, 435–442.

Nigam, K., and Ghani, R. 2000. Analyzing the effectiveness and applicability of co-training. In *In Workshop on information and knowledge management*.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. *CoRR* abs/1802.05365.

Plank, B.; Søgaard, A.; and Goldberg, Y. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. *CoRR* abs/1604.05529.

Sagae, K. 2009. Analysis of discourse structure with syntactic dependencies and data-driven shift-reduce parsing. In *Proceedings of the 11th International Conference on Parsing Technologies*, IWPT '09, 81–84. Stroudsburg, USA: ACL.

Straka, M. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, 197–207. Brussels, Belgium: ACL.

Xu, C.; Tao, D.; and Xu, C. 2013. A survey on multi-view learning. *CoRR* abs/1304.5634.

Yu, S.; Krishnapuram, B.; Rosales, R.; and Rao, R. B. 2011. Bayesian co-training. *Journal of Machine Learning Research* 12(Sep):2649–2680.

Zeman, D., et al. 2018. Universal Dependencies 2.2 – CoNLL 2018 shared task development and test data. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University, Prague, http://hdl.handle.net/11234/1-2184.

Zeman, D.; Hajič, J.; Popel, M.; Potthast, M.; Straka, M.; Ginter, F.; Nivre, J.; and Petrov, S. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, 1–21. Brussels, Belgium: ACL.

Zhao, J.; Xie, X.; Xu, X.; and Sun, S. 2017. Multi-view learning overview: Recent progress and new challenges. *Information Fusion* 38:43–54.