Transductive Learning of Neural Language Models for Syntactic and Semantic Analysis

Hiroki Ouchi\(^1,2\) Jun Suzuki\(^1,2\) Kentaro Inui\(^2,1\)
\(^1\) RIKEN Center for Advanced Intelligence Project \(^2\) Tohoku University
hiroki.ouchi@riken.jp, \{jun.suzuki, inui\}@ecei.tohoku.ac.jp

Abstract

In transductive learning, an unlabeled test set is used for model training. While this setting deviates from the common assumption of a completely unseen test set, it is applicable in many real-world scenarios, where the texts to be processed are known in advance. However, despite its practical advantages, transductive learning is underexplored in natural language processing. Here, we conduct an empirical study of transductive learning for neural models and demonstrate its utility in syntactic and semantic tasks. Specifically, we fine-tune language models (LMs) on an unlabeled test set to obtain test-set-specific word representations. Through extensive experiments, we demonstrate that despite its simplicity, transductive LM fine-tuning consistently improves state-of-the-art neural models in both in-domain and out-of-domain settings.

1 Introduction

In supervised learning, a model is trained on a training set and its generalization performance is evaluated on an unseen test set. In this setting, the model has no access to the test set during training. However, the assumption of a completely unseen test set is not always necessary. In many cases, certain aspects of the test set are already known at training time. For example, a company may want to annotate a large number of existing documents automatically (Section 3). In such a scenario, the texts to be processed are known in advance, and using the model trained on the texts themselves to process them can be more efficient. Using an unlabeled test set in this way is the key idea behind transductive learning.

In transductive learning (Vapnik, 1998), an unlabeled test set is given in the training phase. That is, the inputs of the test set, i.e., the raw texts, can be used during training, but the labels are never used. In the test phase, the trained model is evaluated on the same test set. Despite its practical advantages, transductive learning has received little attention in natural language processing (NLP). After the pioneering work of Joachims (1999), who proposed a transductive support vector machine for text classification, transductive methods for linear models have been investigated in only a few tasks, such as lexical acquisition (Duh and Kirchhoff, 2006) and machine translation (Ueffing et al., 2007). In particular, transductive learning with neural networks is underexplored.

Here, we investigate the impact of transductive learning on state-of-the-art neural models in syntactic and semantic tasks, namely syntactic chunking and semantic role labeling (SRL). Specifically, inspired by recent findings that language model (LM)-based word representations yield large performance improvement (Devlin et al., 2019), we fine-tune Embeddings from Language Models (ELMo) (Peters et al., 2018) on an unlabeled test set and use them in each task-specific model. Typically, LMs are trained on a large-scale corpus whose word distributions are different from the test set. By contrast, transductive learning allows us to fit LMs directly to the distributions of the test set. Our experiments show the effectiveness of transductive LM fine-tuning.

In summary, our main contributions are:

- This work is the first to introduce an LM fine-tuning method to transductive learning\(^1\).
- Through extensive experiments in both in-domain and out-of-domain settings, we demonstrate that transductive LM fine-tuning consistently improves state-of-the-art neural models in syntactic and semantic tasks.

---

\(^1\)Our code and scripts are publicly available at https://github.com/hiroki13/transductive-language-models.
2 Related Work

Transductive learning. Vapnik advocated and formalized transductive learning (Vapnik, 1998; Gammerman et al., 1998), which has been applied to text classification (Joachims, 1999; Ifrim and Weikum, 2006) and image processing (Bruzzone et al., 2006; Sener et al., 2016; Liu et al., 2019).

Although some studies have presented transductive methods for linear models (Duh and Kirchhoff, 2006; Ueffing et al., 2007; Chen et al., 2008; Alexandrescu and Kirchhoff, 2009), transductive methods for neural models are underexplored in NLP.

Unsupervised domain adaptation. Transductive learning is related to unsupervised domain adaptation, in which models are adapted to a target domain by using unlabeled target domain texts (Ben-David et al., 2010; Shi and Sha, 2012). This setting does not allow models to access the test set, which is the main difference between unsupervised domain adaptation and transductive learning. Various unsupervised adaptation methods have been proposed for linear models (Blitzer et al., 2006; Jiang and Zhai, 2007; Tsuboi et al., 2009; Søgaard, 2013). In the context of neural models, adversarial domain adaptation (Ganin and Lempitsky, 2015; Ganin et al., 2016; Guo et al., 2018), importance weighting (Wang et al., 2017), structural correspondence learning (Ziser and Reichart, 2017), self/tri/co-training (Saito et al., 2017; Ruder and Plank, 2018), and other techniques orthogonal to transductive LM fine-tuning have been applied successfully in unsupervised domain adaptation\(^2\). Integrating these methods with transductive LM fine-tuning is an interesting direction for future research.

LM-based word representations. Recently, LM-based word representations pre-trained on unlabeled data have gained considerable attention (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019). The most related method to ours is Universal Language Model Fine-tuning (ULMFiT), which pre-trains an LM on a large general-domain corpus and fine-tunes it on the target task (Howard and Ruder, 2018). Inspired by these studies, we introduce LM-based word representation in transductive learning.

\(^2\)Feature augmentation is considered a supervised domain adaptation method (Daume III, 2007; Kim et al., 2016).

3 Neural Transductive Learning

Motivation. Suppose that a company has received a vast amount of customer reviews and wants to automatically process these reviews more accurately, even if it takes some time. For this purpose, they do not have to build a model that works well on new unseen reviews. Instead, they want a model that works well on only the reviews in hand. In this situation, using these reviews themselves to train a model can be more efficient. This is the key motivation for developing effective and practical transductive learning methods. Toward this goal, we develop transductive methods for state-of-the-art neural models.

Problem formulation. In the training phase, a training set \(D^{train} = \{(X_i^{train}, Y_i^{train})\}_{i=1}^{N^{train}}\) and an unlabeled test set \(D^{test} = \{(X_i^{test})\}_{i=1}^{N^{test}}\) are used for model training, where \(X_i\) is an input, e.g., a sentence, and \(Y_i\) represents target labels, e.g., labels from a set of syntactic or semantic annotations. In the test phase, the trained model is used for predicting labels and is evaluated on the same test set \(D^{test}\).

Method. We present a simple transductive method for neural models. Specifically, we fine-tune an LM on an unlabeled test set. Figure 1 illustrates the training procedure that consists of the following steps: (1) LM pre-training, (2) Transductive LM fine-tuning and (3) task-specific model training. We first train an LM on a large-scale unlabeled corpus \(D^{large}\) and then fine-tune the LM on an unlabeled test set \(D^{test}\). Finally, we use the fine-tuned LM as the embedding layer of
Each task-specific model and train the model on a training set $D_{\text{train}}$.

$$
\Theta' \leftarrow \arg\min_{\Theta} L_{\text{lm}}(\Theta | D_{\text{large}}), \quad (1)
$$

$$
\Theta'' \leftarrow \arg\min_{\Theta} L_{\text{lm}}(\Theta | D_{\text{test}}), \quad (2)
$$

$$
\Phi' \leftarrow \arg\min_{\Phi} L_{\text{task}}(\Phi | \Theta', D_{\text{train}}). \quad (3)
$$

Here, $L_{\text{lm}}$ and $L_{\text{task}}$ are the loss functions for an LM and task-specific model, respectively. In the LM pre-training and fine-tuning phases (Eqs. 1 and 2), we first train the initial LM parameters $\Theta$ and then fine-tune the pre-trained parameters $\Theta'$. In the task-specific training phase (Eq. 3), we fix the fine-tuned LM parameters $\Theta''$ used for the embedding layer of a task-specific model, and train only the task-specific model parameters $\Phi$.

4 Experiments

Tasks. To investigate the effectiveness of transductive LM fine-tuning for syntactic and semantic analysis, we conduct experiments in syntactic chunking (Ramshaw and Marcus, 1999; Sang and Buchholz, 2000; Ponvert et al., 2011) and SRL (Gildea and Jurafsky, 2002; Palmer et al., 2005; Carreras and Márquez, 2005). The goal of syntactic chunking is to divide a sentence into non-overlapping phrases that consist of syntactically related words. The goal of SRL is to identify semantic arguments for each predicate. For example, consider the following sentence:

```
The man kept a cat
```

In syntactic chunking, given the input sentence, systems have to recognize “The man” and “a cat” as noun phrases (NP). In SRL, given the input sentence and the target predicate “kept”, systems have to recognize “The man” as the A0 argument and “a cat” as the A1 argument. For syntactic chunking, we adopted the experimental protocol by Ponvert et al. (2011) and for SRL, we followed Ouchi et al. (2018) (details in Appendix A).

Datasets. We perform experiments using the CoNLL-2012 dataset. To investigate the performances under in-domain and out-of-domain settings, we use each of the seven domains in the CoNLL-2012 dataset. Table 1 shows the data statistics. Each test set contains at most 2,000 sentences. Compared with previous studies, such as Xiao and Guo (2013) that used 570,000 sentences as unlabeled data for unsupervised domain adaptation of syntactic chunking, our transductive experiments can be regarded as a low-resource adaptation setting. As a large-scale unlabeled raw corpus for LM training, we use the 1B word benchmark corpus (Chelba et al., 2013).

Model setup. We use ELMo (Peters et al., 2018) as an LM. For syntactic chunking, we use a variant of the Reconciled Span Parser (Joshi et al., 2018). For SRL, we use the span selection model (BiLSTM-Span model) (Ouchi et al., 2018). Each model is trained on a source domain training set and was evaluated on a target domain test set. The development set is also the source domain, and it is used for hyperparameter tuning. Consider the case where NW $\rightarrow$ BC, i.e., the source domain is the newswire NW and the target domain is the broadcast conversation BC. We first train ELMo on the large-scale raw corpus (one billion word benchmark corpus) and fine-tune it on the BC test set. We then train syntactic and semantic models that use the fine-tuned ELMo on the NW training set. During the task-specific model training, we freeze the fine-tuned ELMo. We select hyperparameters by using the NW development set. Finally, we evaluate the trained model on the BC test set. In the same way, we conduct training and evaluation for each domain pair.

3In our experiments (Section 4), both losses were given by the negative log-likelihood (Appendix A).

4This paper addresses span-based, PropBank-style SRL. Detailed descriptions on other lines of SRL research (e.g. dependency-based SRL and FrameNet-based SRL) can be found in Baker et al. (1998); Das et al. (2014); Surdeanu et al. (2008); Hajić et al. (2009).

5We used the version of OntoNotes downloaded at: http://cemantix.org/data/ontonotes.html.


7All models and hyperparameters are described in Appendices B, C, and D.
ing LMs directly to each test set is more effective than adapting them to target domain unlabeled data. Here, we investigate whether adapting LMs directly to the word distributions of labeled test set can be used for training, it is possible that adapting LMs directly to target domain unlabeled data. Similarly to our transductive method shown in Figure 1, we first train LMs on the large-scale unlabeled corpus (the 1B word benchmark corpus) and then fine-tune them on the unlabeled target domain unlabeled data. In addition, we control the sizes of the target domain unlabeled data and test sets. That is, we use the same number of sentences in the unlabeled data of each target domain as in each test set. Table 3 shows the F1 scores averaged across all the target domains. The transductive models (T) consistently outperformed the domain-adapted models (CU). This demonstrates that adapting LMs directly to test sets is more effective than adapting them to target domain unlabeled data.

Table 3: Performance comparison between LM fine-tuning on target domain unlabeled data of the same size as each test set, “Controlled Unlabeled data (CU),” and transductive LM fine-tuning on each test set (T). Cells show the F1 scores averaged across the target domains.

### 5 Analysis

#### Comparison between unsupervised domain adaptation and transduction.

In unsupervised domain adaptation, target domain unlabeled data (the texts whose domain is the same as that of a test set) is used for adaptation. Although the domain is identical between target domain data and a test set, their word distributions are somewhat different. In transductive learning, because an unlabeled test set can be used for training, it is possible to adapt LMs directly to the word distributions of the test set. Here, we investigate whether adapting LMs directly to each test set is more effective than adapting LMs to each target domain unlabeled data. Similarly to our transductive method shown in Figure 1, we first train LMs on the large-scale unlabeled corpus (the 1B word benchmark corpus) and then fine-tune them on the unlabeled target domain unlabeled data. In addition, we control the sizes of the target domain unlabeled data and test sets. That is, we use the same number of sentences in the unlabeled data of each target domain as in each test set. Table 3 shows the F1 scores averaged across all the target domains. The transductive models (T) consistently outperformed the domain-adapted models (CU). This demonstrates that adapting LMs directly to test sets is more effective than adapting them to target domain unlabeled data.

<table>
<thead>
<tr>
<th>src → tgt</th>
<th>BC</th>
<th>BN</th>
<th>MZ</th>
<th>NW</th>
<th>PT</th>
<th>TC</th>
<th>WB</th>
<th>Averaged F1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Syntactic Chunking</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BC</td>
<td>93.0</td>
<td>93.5</td>
<td>92.9</td>
<td>93.0</td>
<td>90.0</td>
<td>90.6</td>
<td>88.1</td>
<td>88.7</td>
</tr>
<tr>
<td>BN</td>
<td>92.5</td>
<td>93.0</td>
<td>94.7</td>
<td>95.0</td>
<td>91.2</td>
<td>91.4</td>
<td>90.0</td>
<td>90.6</td>
</tr>
<tr>
<td>MZ</td>
<td>89.9</td>
<td>90.8</td>
<td>91.6</td>
<td>92.3</td>
<td>92.5</td>
<td>89.2</td>
<td>90.0</td>
<td>93.4/94.3</td>
</tr>
<tr>
<td>NW</td>
<td>91.4</td>
<td>92.0</td>
<td>93.7</td>
<td>93.9</td>
<td>92.2</td>
<td>92.6</td>
<td>94.2</td>
<td>94.5</td>
</tr>
<tr>
<td>PT</td>
<td>87.1</td>
<td>88.2</td>
<td>86.9</td>
<td>87.5</td>
<td>85.6</td>
<td>86.9</td>
<td>81.0</td>
<td>82.7</td>
</tr>
<tr>
<td>TC</td>
<td>87.3</td>
<td>88.2</td>
<td>87.2</td>
<td>87.5</td>
<td>84.1</td>
<td>85.4</td>
<td>80.8</td>
<td>82.3</td>
</tr>
<tr>
<td>WB</td>
<td>91.8</td>
<td>92.3</td>
<td>93.4</td>
<td>93.7</td>
<td>91.7</td>
<td>92.2</td>
<td>91.0</td>
<td>91.5</td>
</tr>
<tr>
<td><strong>Semantic Role Labeling</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BC</td>
<td>83.3</td>
<td>83.9</td>
<td>78.9</td>
<td>79.3</td>
<td>74.2</td>
<td>74.8</td>
<td>71.0</td>
<td>73.4</td>
</tr>
<tr>
<td>BN</td>
<td>80.3</td>
<td>81.2</td>
<td>83.3</td>
<td>83.5</td>
<td>76.5</td>
<td>77.4</td>
<td>75.0</td>
<td>75.7</td>
</tr>
<tr>
<td>MZ</td>
<td>76.4</td>
<td>77.5</td>
<td>76.6</td>
<td>77.3</td>
<td>80.2</td>
<td>80.6</td>
<td>73.8</td>
<td>74.8</td>
</tr>
<tr>
<td>NW</td>
<td>79.2</td>
<td>80.1</td>
<td>79.8</td>
<td>80.0</td>
<td>79.5</td>
<td>80.0</td>
<td>83.8</td>
<td>84.4</td>
</tr>
<tr>
<td>PT</td>
<td>71.2</td>
<td>72.1</td>
<td>67.4</td>
<td>67.8</td>
<td>66.6</td>
<td>68.0</td>
<td>64.7</td>
<td>66.0</td>
</tr>
<tr>
<td>TC</td>
<td>73.8</td>
<td>74.1</td>
<td>67.6</td>
<td>67.8</td>
<td>64.5</td>
<td>64.9</td>
<td>59.2</td>
<td>60.2</td>
</tr>
<tr>
<td>WB</td>
<td>74.1</td>
<td>74.4</td>
<td>71.7</td>
<td>72.4</td>
<td>72.0</td>
<td>72.8</td>
<td>71.4</td>
<td>72.0</td>
</tr>
</tbody>
</table>

Table 2: Main results under cross-domain settings, src ("source", training set) → tgt ("target", test set). Cells show the F1 scores of the baseline model (before the slash) and the transductive model (after the slash). Column “Averaged F1” represents the F1 scores averaged across the target domains. Domains are as follows: BC = Broadcast Conversation, BN = Broadcast News, MZ = Magazine, NW = Newswire, PT = New Testament, TC = Telephone Conversation, and WB = Weblogs and Newsgroups.

Table 3: Performance comparison between LM fine-tuning on target domain unlabeled data of the same size as each test set, “Controlled Unlabeled data (CU),” and transductive LM fine-tuning on each test set (T). Cells show the F1 scores averaged across the target domains.
Table 4: Performance comparison between LM fine-tuning on target domain unlabeled data (U) and on the combination of the unlabeled data and test sets (U + T). Cells show the F1 scores averaged across the target domains.

<table>
<thead>
<tr>
<th>Domain</th>
<th>CoNLL-2000</th>
<th>CoNLL-2005</th>
<th>CoNLL-2012</th>
</tr>
</thead>
<tbody>
<tr>
<td>BC</td>
<td>90.5</td>
<td>91.0</td>
<td>79.0</td>
</tr>
<tr>
<td>BN</td>
<td>91.3</td>
<td>91.6</td>
<td>80.1</td>
</tr>
<tr>
<td>MZ</td>
<td>90.2</td>
<td>90.6</td>
<td>78.3</td>
</tr>
<tr>
<td>NW</td>
<td>92.1</td>
<td>92.5</td>
<td>81.5</td>
</tr>
<tr>
<td>PT</td>
<td>87.3</td>
<td>87.7</td>
<td>73.6</td>
</tr>
<tr>
<td>TC</td>
<td>87.2</td>
<td>87.6</td>
<td>71.4</td>
</tr>
<tr>
<td>WB</td>
<td>91.8</td>
<td>92.2</td>
<td>76.8</td>
</tr>
</tbody>
</table>

Table 5: Standard benchmark results. Cells show the F1 scores on each test set. The CoNLL-2000 and CoNLL-2005/2012 datasets are used for syntactic chunking and SRL, respectively. Results of the transductive models (TRANS) marked with * are statistically significant compared to the baselines (BASE) using the permutation test (p < 0.05).

Combination of unsupervised domain adaptation and transduction. In real-world situations, large-scale unlabeled data of target domains is sometimes available. In such cases, LMs can be trained on both the target domain unlabeled data and the test sets. Here, we investigate the effectiveness of using both datasets. Table 4 shows the F1 scores averaged across all the target domains. Fine-tuning the LMs on the target domain unlabeled data as well as each test set (U + T) showed better performance than fine-tuning them only on the target domain unlabeled data (U). This combination of transduction with unsupervised domain adaptation further improves performance.

Effects in standard benchmarks. Some studies indicated that when promising new techniques are only evaluated on very basic models, determining how much (if any) improvement will carry over to stronger models can be difficult (Denkowski and Neubig, 2017; Suzuki et al., 2018). Motivated by such studies, we provide the results in standard benchmark settings. For syntactic chunking, we use the CoNLL-2000 dataset (Sang and Buchholz, 2000) and follow the standard experimental protocol (Hashimoto et al., 2017). For SRL, we use the CoNLL-2005 (Carreras and Marquez, 2005) and CoNLL-2012 datasets (Pradhan et al., 2012) and follow the standard experimental protocol (Ouchi et al., 2018). Table 5 shows the F1 scores of our models and those of existing models. The results of the baseline model were comparable with those of the state-of-the-art models, and the transductive model consistently outperformed the baseline model. Note that we cannot fairly compare the transductive and existing models due to the difference in settings. These results, however, demonstrate that transductive LM fine-tuning improves state-of-the-art chunking and SRL models.

6 Conclusion

In this study, we investigated the impact of transductive learning on state-of-the-art neural models in syntactic and semantic tasks. Specifically, we fine-tuned an LM on an unlabeled test set. Through extensive experiments, we demonstrated that, despite its simplicity, transductive LM fine-tuning contributes to consistent performance improvement of state-of-the-art syntactic and semantic models in cross-domain settings. One interesting line of future work is to explore effective transductive methods for task-dependent (neural) layers. For instance, as some unsupervised domain adaptation methods can be applied to transductive learning, integrating them with transductive LM fine-tuning may further improve their performance. Another line of our future work is to apply these transductive methods to various NLP tasks and investigate their performance.

Acknowledgments

This work was partially supported by JSPS KAKENHI Grant Number JP19H04162 and JP19K20351. We would like to thank Benjamin Heinezling, Ana Brassard, Sosuke Kobayashi, Hitomi Yanaka, and the anonymous reviewers for their insightful comments.

While the improvements in SRL were statistically significant compared to the baseline, the improvement in syntactic chunking was not. One reason for this is that the F1 score of the baseline in syntactic chunking is already high and there is less room for improvement. Since Clark et al. (2018) achieved 97.0 F1 with multi-task learning, missing information for further improvement might be derived from other tasks.
References


Zuchao Li, Shexia He, Hai Zhano, Yiqing Zhang, Zhusongheng Zhang, Zhou Xi, and Xiang Zhou. 2019. Dependency or span, end-to-end uniform semantic role labeling. In Proceedings of AAAI.


