# Out-of-Domain Discourse Dependency Parsing via Bootstrapping: An Empirical Analysis on Its Effectiveness and Limitation

**Noriki Nishida and Yuji Matsumoto**
RIKEN Center for Advanced Intelligence Project, Japan
{noriki.nishida, yuji.matsumoto}@riken.jp

## Abstract

Discourse parsing has been studied for decades. However, it still remains challenging to utilize discourse parsing for real-world appli- cations because the parsing accuracy degrades significantly on out-of-domain text. In this paper, we report and discuss the effectiveness and limitations of bootstrapping methods for adapting modern BERT-based discourse dependency parsers to out-of-domain text without relying on additional human supervision.Specifically, we investigate self-training, cotraining, tri-training, and asymmetric tri-training of graph-based and transition-based discourse dependency parsing models, as well as confidence measures and sample selection criteria in two adaptation scenarios: monologue adaptation between scientific disciplines and dialogue genre adaptation. We also release COVID-19 Discourse Dependency Treebank (COVID19-DTB), a new manually annotated resource for discourse dependency parsing of biomedical paper abstracts. The experimental results show that bootstrapping is significantly and consistently effective for unsupervised domain adaptation of discourse dependency parsing, but the low coverage of accurately predicted pseudo labels is a bottleneck for further improvement. We show that active learning can mitigate this limitation.

## 1 Introduction

Discourse parsing aims to uncover structural organization of text, which is useful in Natural Language Processing (NLP) applications such as document summarization (Louis et al., 2010; Hirao et al., 2013; Yoshida et al., 2014; Bhatia et al., 2015; Durrett et al., 2016; Xu et al., 2020), text categorization (Ji and Smith, 2017; Ferracane et al., 2017), question answering (Verberne et al., 2007; Jansen et al., 2014), and information extraction (Quirk and Poon, 2017). In particular, dependency-style representation of discourse structure has been studied intensively in recent years (Asher and Lascarides, 2003; Hirao et al., 2013; Li et al., 2014b; Morey et al., 2018; Hu et al., 2019; Shi and Huang, 2019). Figure 1 shows an example of discourse depen- dency structure, which is recorded in COVID-19 Discourse Dependency Treebank (COVID19-DTB), a new manually annotated resource for discourse dependency parsing of biomedical abstracts. State-of-the-art discourse dependency parsers are generally trained on a manually annotated treebank, which is available in a limited number of domains, such as RST-DT (Carlson et al., 2001) for news articles, SciDTB (Yang and Li, 2018) for NLP abstracts, and STAC (Asher et al., 2016) and Molweni (Li et al., 2020) for multi-party dialogues. However, when the parser is applied directly to out-of-domain documents, the parsing accuracy degrades significantly due to the domain shift problem. In fact, we normally face this issue in the real world because human supervision is generally scarce and expensive to obtain in the domain of interest. Unsupervised Domain Adaptation (UDA) aims to adapt a model trained on a source domain, where a limited amount of labeled data is available, to a target domain, where only unlabeled data is available. Bootstrapping (or pseudo labeling) has been shown to be effective for the UDA problem of syntactic parsing (Steedman et al(Louis et al., 2010)., 2003b,a; Reichart and Rappoport, 2007; Søaard and Rishøj, 2010; Weiss et al., 2015). In bootstrapping for syntactic parsing, we first train a model on the labeled source sentences, the model is used to give pseudo labels (i.e., parse trees) to unlabeled target sentences, and then the model is retrained on the manually and automatically labeled sentences. On the contrary, despite the significant progress achieved in discourse parsing so far (Li

et al., 2014b; Ji and Eisenstein, 2014; Joty et al., 2015; Perret et al., 2016; Wang et al., 2017; Kobayashi et al., 2020; Koto et al., 2021), boot-

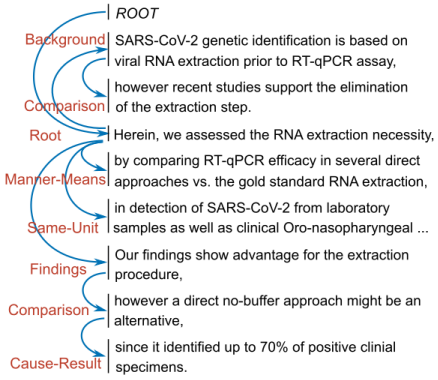


Figure 1: An example of discourse dependency structure for a COVID-19 related biomedical paper abstract (Israeli et al., 2020), which we manually annotated for our new dataset.

strapping for the UDA problem of discourse parsing is still not well understood. Jiang et al. (2016) and Kobayashi et al. (2021) explored how to enrich the labeled dataset using bootstrapping methods; however, their studies are limited to the in-domain setup, where the labeled and unlabeled datasets are derived from the same domain. In contrast to these studies, we focus on the more realistic and challenging scenario, namely, out-of-domain discourse parsing, where the quality and diversity of the pseudo-labeled dataset become more crucial for performance enhancement. In this paper, we perform a series of analyses of various bootstrapping methods in UDA of modern BERT-based discourse dependency parsers and report the effectiveness and limitations of these approaches. Figure 2 shows an overview of our bootstrapping system. Specifically, we investigate self-training (Yarowsky, 1995), co-training (Blum and Mitchell, 1998; Zhou and Goldman, 2004), tri-training (Zhou and Li, 2005), and asymmet- ric tri-training (Saito et al., 2017) of graph-based and transition-based discourse dependency pars- ing models, as well as confidence measures and sample selection criteria in two adaptation sce- narios: monologue adaptation between scientific disciplines and dialogue genre adaptation. We show that bootstrapping improves out-of-domain discourse dependency parsing significantly and consistently across different adaptation setups.

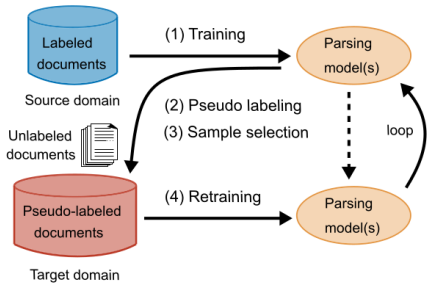


Figure 2: An overview of our bootstrapping system for unsupervised domain adaptation of discourse depen- dency parsing.

## 2 Related Works

Various discourse parsing models have been proposed in the past decades. For constituency- style discourse structure like RST (Mann and Thompson, 1988), the parsing models can be cate- gorized into the chart-based approach (Joty et al., 2013; Joty et al., 2015; Li et al., 2014a, 2016a), which finds the globally optimal tree using an efficient algorithm like dynamic programming, or the transition-based (or sequential) approach (Marcu, 1999; Sagae, 2009; Hernault et al., 2010b; Feng and Hirst, 2014; Ji and Eisenstein, 2014; Wang et al., 2017; Kobayashi et al., 2020; Zhang et al., 2020; Koto et al., 2021), which builds a tree incrementally by performing a series of decisions. For dependency-style discourse structure like the RST variants (Hirao et al., 2013; Li et al., 2014b; Morey et al., 2018) or Segmented Discourse Representation Theory (Asher and Lascarides, 2003), the models can also be categorized into the graph-based approach (Li et al., 2014b; Yoshida et al., 2014; Afantenos et al., 2015; Perret et al., 2016) or the transition-based (sequential) approach (Muller et al., 2012; Hu et al., 2019; Shi and Huang, 2019). Recently, pre-trained transformer encoders such as BERT (Devlin et al., 2019) and SpanBERT (Joshi et al., 2019) have been shown to greatly improve discourse parsing accuracy (Guz and Carenini, 2020; Koto et al., 2021). In this paper, we are not aiming at developing novel parsing models. Instead, we aim to investigate the effectiveness and limitations of bootstrapping methods for adapting the modern BERT-based discourse parsers. Manually annotated discourse treebanks are sig- nificantly scarce, and their domains are limited. For example, the most popular discourse tree- bank, RST-DT (Carlson et al., 2001), contains only 385

labeled documents in total. To address the lack of large-scale labeled data, a number of semi-supervised, weakly supervised, and un- supervised techniques have been proposed in the discourse parsing literature. Hernault et al. (2010a) proposed a semi-supervised method that utilizes unlabeled documents to expand feature vectors in SVM classifiers in order to achieve better generalization for infrequent discourse relations. Liu and Lapata (2018) and Huber and Carenini (2019) proposed to exploit document-level class labels (e.g., sentiment) as distant supervision to in- duce discourse dependency structures from neural attention weights. Badene et al. (2019a,b) investigated a data programming paradigm (Ratner et al., 2016), which uses rule-based labeling functions to automatically annotate unlabeled documents and trains a generative model on the weakly supervised data. Kobayashi et al. (2019) and Nishida and Nakayama (2020) proposed fully unsupervised discourse constituency parsers, which can produce only tree skeletons and rely strongly on pre-trained word embeddings or human prior knowledge on document structure. Technically most similar to our work, Jiang et al. (2016) and Kobayashi et al. (2021) proposed to enlarge the training dataset using a combination of multiple parsing models. Jiang et al. (2016) used co-training for enlarging the RST-DT train- ing set with 2,000 Wall Street Journal articles, with a focus on improving classification accu- racy on infrequent discourse relations. Kobayashi et al. (2021) proposed to exploit discourse sub- trees that are agreed by two different models for enlarging the RST-DT training set. Interestingly, their proposed methods improved the classifica- tion accuracy especially for infrequent discourse relations. These studies mainly assume the in-domain sce- nario and focus on enlarging the labeled set (e.g., RST-DT training set) using in-domain unlabeled documents, and the system evaluation is generally performed on the same domain with the original labeled set (e.g., RST-DT test set). In this paper, instead, we particularly focus on the UDA sce- nario, where the goal is to parse the target-domain documents accurately without relying on human supervision in the target domain. We believe this research direction is important for developing us- able discourse parsers, because a target domain to which one would like to apply a discourse parser is normally different from the domains/genres of ex- isting corpora, and manually annotated resources are rarely available in most domains/genres.

# 3 Method

## 3.1 Problem Formulation

The input is a document represented as a sequence of clause-level (in single-authored text) or utterance-level (in multi-party dialogues) spans called Elementary Discourse Units (EDUs). Our goal is to derive a discourse dependency structure, $y = (h, d, r) \mid 0 \leq h \leq n, 1 \leq d \leq n, r \in R$, given the input EDUs, $x = e_0, e_1, \ldots, e_n$, which is analogous to syntactic dependency structure. A discourse dependency, $(h, d, r)$, represents that the $d$-th EDU (called dependent) relates to the $h$-th EDU (called head) directly with the discourse relation $r \in R$. Each EDU except for the root node, $e_0$, has a single head.

In this paper, we assume that we have a limited number of labeled documents in the source domain, while a large collection of unlabeled documents is available in the target domain. In particular, we assume that the source and target domains have different data distributions lexically or rhetorically (e.g., vocabulary, document length, and discourse relation distributions), but the domains share the same annotation scheme (e.g., definition of discourse relation classes). Our task is to adapt a parsing model (or models) trained in the source domain to the target domain using the unlabeled target data.

## 3.2 Bootstraping

The aim of this paper is to investigate the effectiveness and limitations of various bootstrapping methods in UDA of modern BERT-based discourse dependency parsers. We show the overall flow of the bootstrapping methods in Figure 2. Ini- tially we have a small set of labeled documents, Ls , in the source domain, and a large collec- tion of unlabeled documents, U t , in the target domain. Then the bootstrapping procedure works as follows: (1) Train initial models on Ls = (xs , y s ). (2) Parse unlabeled documents xt U t using the current model f , such as, f : U t $\rightarrow$ Lt = (xt , f (xt )).3 , 4 (3) Measure the confidence scores of the pseudo-labeled data and select a subset, Lt Lt , that is expected to be reliable and useful. (4) Retrain the models on Ls Lt for several epochs (set to 3 in this work). Steps (2)-(4) loop

for many rounds until a predefined stopping criterion is met. Bootstrapping can be interpreted as a methodol- ogy where teachers generate pseudo supervision for students, and the students learn the task on it. Existing bootstrapping methods vary depending on how the teacher and student models are used. In this paper, we specifically explore the following bootstrapping methods: self-training (Yarowsky, 1995; McClosky et al., 2006; Reichart and Rappoport, 2007; Suzuki and Isozaki, 2008; Huang and Harper, 2009), co-training (Blum and Mitchell, 1998; Zhou and Goldman, 2004; Steedman et al., 2003b,a), tri-training (Zhou and Li, 2005; Weiss et al., 2015; Ruder and Plank, 2018), and asym- metric tri-training (Saito et al., 2017).

**Self Training** Self-Training (ST) starts with a single model f trained on Ls . The overall proce- dure is the same as the one described above. The single model is both a teacher and a student for itself. Thus, it is difficult for the model to obtain novel knowledge (or supervision) that the model has not learn, and its errors may be amplified by the retraining cycle.

**Co-Training** Co-Training (CT) starts with two parsing models, f1 and f2 , that are expected to have different inductive biases with each other. The two models are pre-trained on the same Ls . In Step 2, each model independently parses the unlabeled documents: U t $\rightarrow$ Lti (i = 1, 2). In Step 3, each of the pseudo-labeled sets are filtered by a selection criterion: Lti $\rightarrow$ Lti . In Step 4, each model fi is retrained on Ls  Ltj (j = i). In CT, the two models teach each other. Thus, each model is the teacher and the student for the other model simultaneously. In contrast to ST, each model can obtain knowledge that it has not yet learned. CT can be viewed as enhancing the agreement between the models.

**Tri-Training(TT)** Tri-Training (TT) consists of three different models, f1 , f2 , and f3 , which are initially trained on the same Ls . In contrast to CT, where the single teacher $fi$ is used to generate pseudo labels $Lti$ for the student $fj(j = i)$, TT uses two teachers, $fi$ and $fj(j = i)$, to generate a pseudo-labeled set $Lti$,j for the remaining student $fk(k = i, j)$. We measure the confidence for the pair of teachers' parse trees, $(yit, yjt)$, using the ratio of agreed dependencies (described in Subsection 3.4), based on which we determine whether or not to include the teachers' predictions in the pseudo-labeled set.

**Asymmetric Tri-Training (AT)** Asymmetric Tri-training (AT) is an extension of TT for UDA. A special domain-specific model f1t is used only for test inference; the other two models, f2 and f3 , are used only to generate pseudo labels $Lt$ . The domain-specific model f1t is retrained on only $Lt$ , while f2 and f3 are retrained on $LsLt$ .

### 3.3 Parsing Models

We employ three types of BERT-based dis- course dependency parsers: (1) A graph-based arc-factored model (McDonald et al., 2005) with a biaffine attention mechanism (Dozat and Manning, 2017), (2) a transition-based shift- reduce model (Nivre, 2004; Chen and Manning, 2014; Kiperwasser and Goldberg, 2016), and (3) the backward variant of the shift-reduce model. EDU Embedding We compute EDU embed- dings using a pretrained Transformer encoder. This manner is common across the three parsing models, though the Transformer parameters are untied and fine-tuned separately. Specifically, we first break down the input document into non- overlapping segments of 512 subtokens, and then encode each segment independently by the Trans- former encoder. Lastly, we compute EDU-level span embeddings as a concatenation of the Trans- former output states at the span endpoints ($wi$ and $wj$ ) and the span-level syntactic head word5 wk , i.e., $[wi; wj; wk]$.

**Arc-Factored Model** Arc-Factored Model (A) is a graph-based dependency parser, which can find the globally optimal dependency structure using dynamic programming. Specifically, we employ the biaffine attention model (Dozat and Manning, 2017) for computing dependency scores $s(h, d) \in R$, and we decode the optimal structure $y^*$ using Eisner Algorithm, such that the tree score $\sum_{(h,d)\in y} s(h, d)$ is maximized. We predict the discourse relation classes for each unlabeled dependency $(h, d) \in y^*$ using another biaffine attention layer and MLP, namely, $r^* = rargmaxP(r \mid h, d)$. To reduce the computational time for inference, we employed the Hierarchical Eisner Algorithm (Zhang et al., 2021), which decodes dependency trees from the sentence level to the paragraph level and then to the whole text level.

**Shift-Reduce Model** Shift-Reduce Model (S) is a transition-based dependency parser, which builds a dependency structure incrementally by executing a series of local actions. Specifically, we employ the arc-standard system proposed by Nivre

(2004), which has a buffer to store the input EDUs to be analyzed and a stack to store the in-progress subtrees and defines the following action classes: SHIFT, RIGHT-ARC-l, and LEFT-ARC-l. We decode the dependency structure y using a greedy search algorithm, i.e., taking the action a that is valid and the most probable at each decision step: $a = argmaxP(a|)$, where denotes the parsing configuration.

**Backward Shift-Reduce Model** We expect that different inductive biases can be introduced by processing the document from the back. As the third model option, we develop a backward variant of the Shift-Reduce Model (B), which processes the input sequence in the reverse order.

### 3.4 Confidence Measures

The key challenge in bootstrapping on out-of- domain data is how to assess the reliability (or usefulness) of the pseudo labels and how to select an error-free and high-coverage subset. We define confidence measures to assess the reliability of the pseudo-labeled data. In Section 3.5, we define selection criteria to filter out unreliable pseudo- labeled data based on their confidence scores.

**Model-based Confidence** For the bootstrap- ping methods that use a single teacher to generate a pseudo-labeled set (i.e., ST, CT), we define the confidence of the teacher model based on predictive probabilities of the decisions used to build a parse tree. A discourse dependency struc- ture consists of a set (or series) of decisions. Therefore, we use the average of the predictive probabilities over the decisions.6 How to calculate the model-based confidence measure $C(x, y)$ depends on the parsing models:

$$C(x,y) = \frac{1}{2n}\sum_{d=1}^{n} P(h|d) + P(r|h,d)\ , \quad (1)$$

where $(h, d, r) \in y$.

where $A(x, y)$ denotes the action and configuration sequence to produce the parse tree y for x.

**Agreement-based Confidence** For the bootstrapping methods that use multiple teachers to generate a pseudo-labeled set (i.e., TT, AT), we use the agreement level between the two teacher models as the confidence for the pseudolabeled data. Specifically, we compute the rate of labeled dependencies agreed between two predicted structures, yi and yj , as follows:

$$C\left(x, y_i, y_j\right) = \frac{1}{n}\sum_{d=1}^{n} 1\left[h_d^i = h_d^j \wedge r_d^i = r_d^j\right]$$

where $1[]$ is the indicator function, and hid and rdi denote the head and the discourse relation class for the dependent d in yi , respectively. It is worth noting that both yi and yj have the same number of dependencies, n. The higher the percentage is, the more correct dependencies are expected to be included.

| COVID19-DTB | SciDTB |
|---|---|
| Root | Root |
| Elaboration | Elaboration, Progression, Summary |
| Comparison | Contrast, Comparison |
| Cause-Result | Cause-Effect, Explain |
| Condition | Condition |
| Temporal | Temporal |
| Joint | Joint |
| Enablement | Enablement |
| Manner-Means | Manner-Means |
| Attribution | Attribution |
| Background | Background |
| Findings | Evaluation |
| Textual-Organization | - |
| Same-Unit | Same-Unit |

Table 1: Discourse relation classes in COVID19-DTB and their correspondences with SciDTB's classes.

### 3.5 Sample Selection Criteria

Inspired by Steedman et al. (2003a), we define two kinds of sample selection criteria, each of which focuses on the reliability (i.e., accuracy) and the usefulness (i.e., training utility) of the data, respectively.

**Rank-above-k** This is a reliability-oriented selection criterion. We keep only the top N × k samples with higher confidence scores, where N is the number of candidate pseudo-labeled data, and k [0.0, 1.0]. Specifically, we first rank the candidate pseudo-labeled data based on the teacher-side confidence scores, and then we select a subset that satisfies $R(x)Nk$ . where $R(x)[1, N]$ denotes the ranking of x.

**Rank-diff-k** This is a utility-oriented selection criterion. In contrast to Rank-above-k , which relies only on the teacher-side confidence, this crite-

| | COVID19-DTB | SciDTB |
|---|---|---|
| Total number of documents | 300 | 1045 (unique: 798) |
| Total number of EDUs | 6005 | 15723 |
| Avg number of EDUs / doc | 20.0 | 15.0 |
| Avg dependency distance | 2.7 | 2.5 |
| Max. dependency distance | 38 | 26 |
| Avg Root position | 6.6 | 3.9 |

Table 2: Dataset statistics for the COVID19-DTB and SciDTB datasets.

rion utilizes both the teacher-side and the student-side confidence scores. This criterion retain the pseudo-labeled data whose relative rank- ing on the teacher side is higher than the relative ranking on the student side by a margin k or more. Specifically, after ranking the candidates indepen- dently for each side, we compute the gap of the relative rankings on the two sides, and then select a subset that meets $Rteacher(x) + kRstudent(x)$.

## 4 COVID19-DTB

We release a new discourse dependency tree- bank for scholarly paper abstracts on COVID-19 and related coronaviruses like SARS and MERS in order to test unsupervised domain adaptation of discourse dependency parsing. We name our new treebank COVID-19 Discourse Dependency Treebank (COVID19-DTB).

### 4.1 Construction

We followed the RST-DT annotation guideline (Carlson and Marcu, 2001) for EDU segmentation. Based on SciDTB and Penn Discourse Treebank (PDTB) (Prasad et al., 2008), we defined 14 discourse relation classes shown in Table 1. We carefully analyzed the annotation data of SciDTB and found that some classes are hard

to discriminate even for humans, which can lead to undesirable inconsistencies in the new dataset. Thus, we have merged some classes, such as Cause-Effect + Explain → Cause-Result. Some classes are also renamed from SciDTB to fit the biomedical domain, such as Evaluation → Findings. First, we sampled 300 abstracts randomly from the 2020 September snapshot of The COVID-19 Open Research Dataset (CORD-19) (Wang et al., 2020), which contains over 500,000 scholarly arti- cles on COVID-19 and related coronaviruses like SARS and MERS. Then, the 300 abstracts were segmented into EDUs manually by the authors. Then, we employed two professional annotators to give gold discourse dependency structures to the 300 abstracts. The annotators were trained using a few examples and a manual guideline, and then they annotated the 300 abstracts independently.7 We divided the results into development and test splits, each of which consists of 150 examples.

### 4.2 Corpus Statistics

Table 2 and Figure 3 show the statistics and the discourse relation distribution of COVID19-DTB. We also show the statistics and the distribution of SciDTB for comparison. We mapped discourse relations in SciDTB to the corresponding classes in COVID19-DTB. We removed the Root relations in computing the proportions. The average number of EDUs per document in each corpus was 20.0 and 15.0, respectively. Although the average dependency distances in the two corpora are almost the same (2.7 vs. 2.5), the maximum dependency distance of COVID19- DTB is significantly longer than that of SciDTB. Furthermore, the average position of Root's direct dependent is located further back in COVID19- DTB (6.6 vs. 3.9). Although the overall discourse relation distributions look similar, the propor- tions of Elaboration and Same-Unit are larger in COVID19-DTB. These differences reflect the fact that biomedical abstracts tend to be longer, have more complex sentences with embedded clauses, and contain more detailed information, suggesting the difficulty of discourse parser adaptation across the two domains.

## 5 Experimental Setup

**Datasets** We evaluated the bootstrapping methods on two UDA scenarios: The first setup was a monologue adaptation between scientific disciplines: NLP and biomedicine (especially on COVID-19), which is actually an important scenario because there is still no text-level discourse treebank on biomedical documents. We used the training split of **SciDTB** (Yang and Li, 2018) as the labeled source dataset, which contains 742

manual discourse dependency structures on the ab- stracts in ACL Anthology. We also used the 2020 September snapshot of **CORD-19** (Wang et al., 2020) as the unlabeled target dataset, which con- tains about 76,000 biomedical abstracts. We used the development and test splits of COVID19-DTB for validation and testing, respectively. The discourse relation labels in the SciDTB training set were mapped to the corresponding classes of **COVID19-DTB**. We mapped Textual- Organization relations in **COVID19-DTB** to Elab- oration, because there is no corresponding class in SciDTB. We also mapped Temporal relations in the two datasets to Condition to reduce the significant class imbalance. The second setup was an adaptation across di- alogue genres, that is, dialogues in a multi-party game and dialogues in Ubuntu Forum. We used the training split of **STAC** (Asher et al., 2016) as the labeled source dataset, which contains 887 manually labeled discourse dependency structures on multi-party dialogues in the game, The Settlers of Catan. We also used the **Ubuntu Dialogue Corpus** (Lowe et al., 2015) as the unlabeled tar- get dataset, which contains dialogues extracted from the Ubuntu chat logs. We retained dialogues with 7-16 utterances and 2-9 speakers. We also removed dialogues with long utterances (more than 20 words). Finally, we obtained approxi- mately 70,000 dialogues. We used the develop- ment and test splits of **Molweni** (Li et al., 2020) for validation and testing. Each split contains 500 manually labeled discourse dependency structures on multi-party dialogues derived from the Ubuntu Dialogue Corpus. The unlabeled target documents in both se- tups were segmented into EDUs using a publicly available EDU segmentation tool (Wang et al., 2018).

**Evaluation** We employed the traditional evaluation metrics in dependency parsing literature, namely, Labeled Attachment Score (LAS) and Unlabeled Attachment Score (UAS). We also used Root Accuracy (RA), which indicates how well a system can identify the most representative EDU in the document (i.e., the dependent of the special root node). **Implementation Details** As the pre-trained transformer encoders, we used SciBERT (Beltagy et al., 2019) and SpanBERT (Joshi et al., 2019) in the first and second adaptation setups, respec- tively. The dimensionality of the MLPs in the arc-factored model and the shift-reduce models are 100 and 128, respectively. We used AdamW and Adam optimizers for optimizing the trans- former's parameters (bert ) and the task-specific parameters (task ), respectively, following Joshi et al. (2019). We first trained the base models on the labeled source dataset using the following hyper-parameters: batch size = 1, learning rate (LR) for bert = 2e5 , LR for task = 1e4 , warmup steps = 2.4K. Then, we ran the bootstrapping meth- ods using the models with: batch size = 1, LR for bert = 2e6 , LR for task = 1e5 , warmup steps = 7K. We trained all approaches for a maximum of 40 epochs. We applied early stopping when the validation LAS does not increase for 10 epochs.

## References

Annie Louis, Aravind K Joshi, and Ani Nenkova. 2010. Discourse indicators for content selection in summaization.

| | | Abstracts | | | |
|---|---|---|---|---|---|
| Dialogues | | | | | |
| Method | Selection | LAS | UAS | RA | LAS |
| UAS | | | | | |
| Source-only (A) | - | 61.3 | 74.8 | $\frac{82.0}{}$ | 29.9 |
| 55.1 | | | | | |
| Source-only (S) | - | $\frac{61.8}{74.5}$ | 78.0 | $\frac{33.2}{6}$ | $\frac{66.1}{55.6}$ |
| Source-only (B) | - | 60.0 | 72.9 | 78.0 | 29.2 |
| 55.7 | | | | | |
| ST (A ← A) | above-0.6 | 65.8 | 78.7 | **88.7** | 34.7 |
| 60.6 | | | | | |
| ST (S ← S) | above-0.6 | 65.3 | 76.9 | 84.7 | 37.9 |
| 67.4 | | | | | |
| CT (A ← S) | above-0.6 | **66.2** | 78.1 | 86.0 | 38.0 |
| 64.8 | | | | | |
| CT (S ← A) | above-0.6 | 66.1 | 78.2 | 86.0 | 39.1 |
| 64.4 | | | | | |
| CT (A ← S) | diff-100 | 66.0 | 78.3 | 88.0 | 38.5 |
| 66.5 | | | | | |
| CT (S ← A) | diff-100 | **66.2** | **78.8** | 84.7 | **39.5** |
| 66.0 | | | | | |
| CT (S ← B) | above-0.6 | 65.3 | 76.8 | 84.0 | 38.1 |
| 67.2 | | | | | |
| CT (B ← S) | above-0.6 | 65.6 | 76.9 | 87.3 | 38.5 |
| 67.4 | | | | | |
| CT (S ← B) | diff-100 | 65.5 | 76.8 | 86.0 | 39.1 |
| 67.5 | | | | | |
| CT (B ← S) | diff-100 | 65.5 | 76.6 | 86.7 | 39.2 |
| **67.7** | | | | | |
| TT (A ← S, B) | above-0.6 | 65.9 | 78.5 | 87.3 | 38.5 |
| 66.6 | | | | | |
| TT (S ← A, B) | above-0.6 | 65.9 | 78.4 | 86.0 | 39.1 |
| 66.7 | | | | | |
| TT (A ← S, B) | diff-100 | 65.4 | 77.4 | 86.7 | 38.6 |
| 66.8 | | | | | |
| TT (S ← A, B) | diff-100 | 65.1 | 77.7 | 87.3 | 38.9 |
| 66.5 | | | | | |
| AT (A ← S, B) | above-0.6 | 64.9 | 77.3 | 85.3 | 36.9 |
| 66.7 | | | | | |
| AT (S ← A, B) | above-0.6 | 65.3 | 77.4 | **88.7** | 38.6 |
| 63.2 | | | | | |
| AT (A ← S, B) | diff-100 | 65.3 | 77.6 | 84.7 | 36.9 |
| 65.7 | | | | | |
| AT (S ← A, B) | diff-100 | 64.6 | 77.6 | 85.3 | 38.2 |
| 61.9 | | | | | |

Table 3: LAS for methods with and without bootstrapping in the two UDA setups. Arrows indicate the teacher and student models: For example, TT (S ← A, B) shows the test performance of the Shift-Reduce Model (S) that is trained with the Arc-Factored Model (A) and the Backward Shift-Reduce Model (B) using Tri-Training (TT). RA is omitted for the dialogue adaptation setup because the accuracy is nearly 100% for most systems.